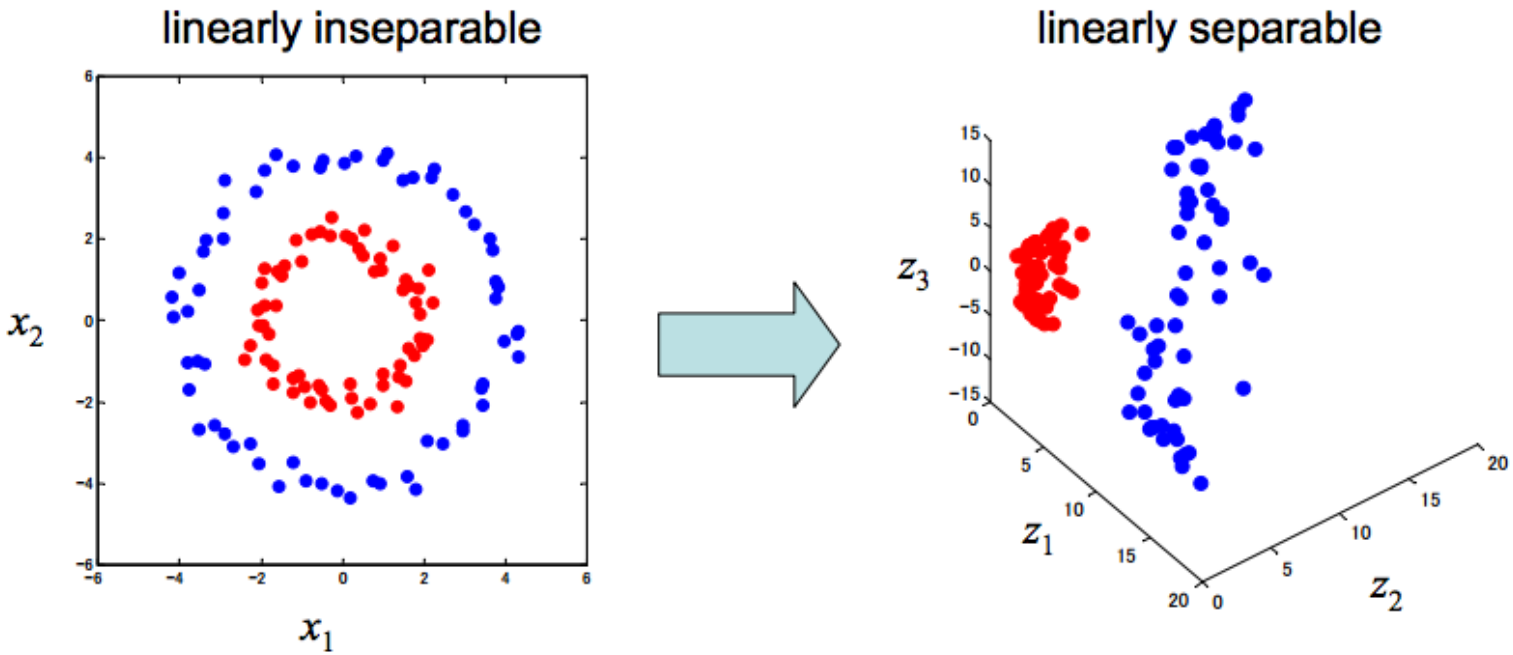
Kernels Methods

Learning scalar-valued and vector-valued functions

Hachem Kadri

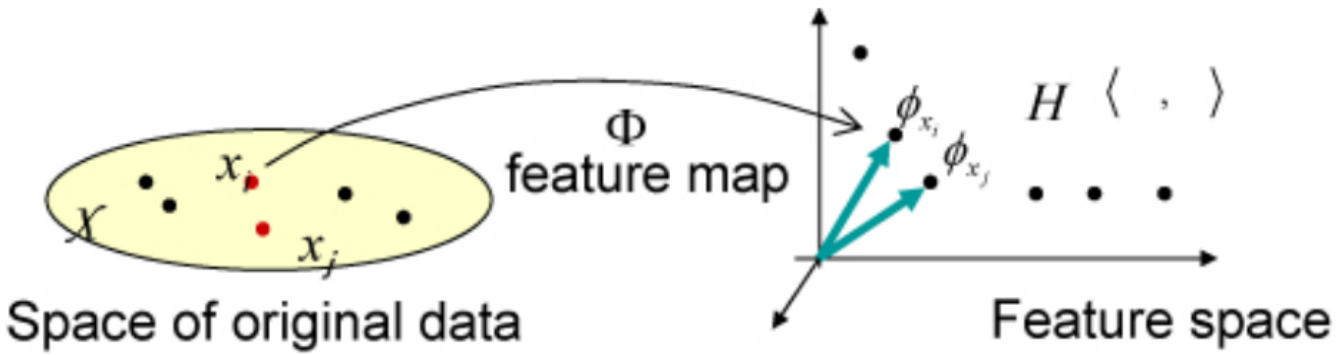
QARMA team
LIS - Université Aix-Marseille

hachem.kadri@lis-lab.fr



$$(x_1, x_2) \mapsto (z_1, z_2, z_3) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

• Noyau - Méthodologie : un moyen systématique pour l'analyse des données en les transformant dans un espace de caractéristique de plus grande dimension



• Appliquer les méthodes linéaires dans l'espace de caractéristiques

$\longrightarrow$ Quel type d'espace peut-être un espace de caractéristique
$\longrightarrow$ L'espace doit intégrer divers types de non-linéarité
$\longrightarrow$ Produit scalaire dans l'espace de caractéristique est primordiale

PART I

From Linear Regression to Kernel Regression

Scalar-valued Kernels

Outline

- **Linear Regression**
  - → Simple Regression
  - → Multiple Regression
  - → Functional learning - scalar outputs
  - → Towards Nonlinear Regression
- **Reproducing Kernels**
  - → Reproducing Kernel Hilbert Space (RKHS)
  - → Positive Kernels
- **Nonlinear Regression with Kernels**
  - → Optimization Problem
  - → Representer Theorem

Linear Regression

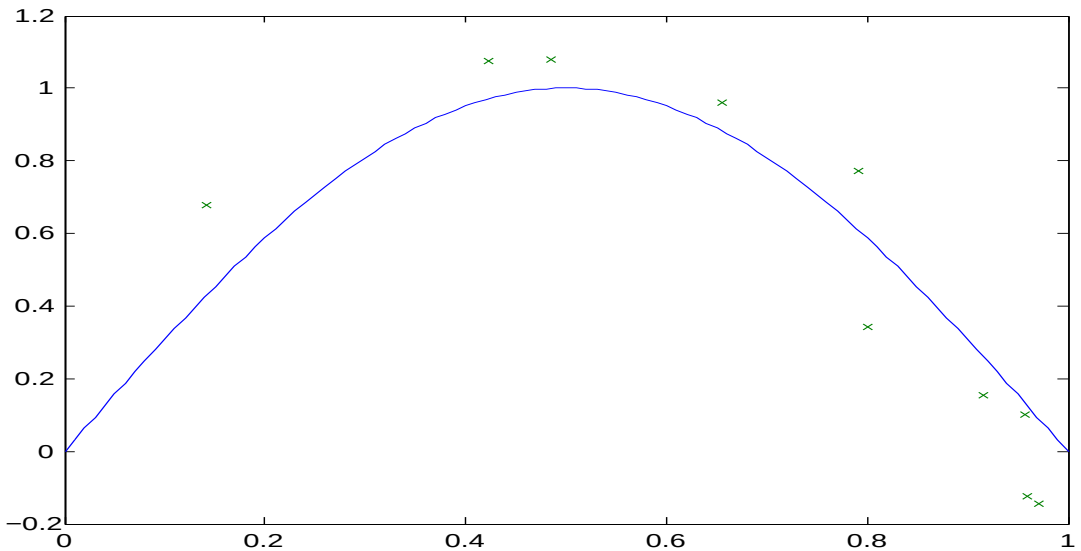
Reference : T. Hastie, R. Tibshirani, J. Friedman (2008).
The Elements of Statistical Learning (2 ed.). New York : Springer.

Problème de régression

On observe des données

$$(x_1, y_1), \ldots, (x_l, y_l) \in \mathbb{R}^n \times \mathbb{R}$$

On cherche à exprimer la dépendance entre x et y par une fonction.



Un exemple : USCrime (L. Wasserman)

- These data are crime-related and demographic statistics for 47 US states in 1960.
- The data were collected from the FBI's Uniform Crime Report and other government agencies to determine how the variable crime rate depends on the other variables measured in the study.

Un exemple : USCrime (suite)

1. R : Crime rate : # of offenses reported to police per million population
2. Age : The number of males of age 14-24 per 1000 population
3. S : Indicator variable for Southern states (0 = No, 1 = Yes) Ed : Mean # of years of schooling x 10 for persons of age 25 or older
4. Ex0 : 1960 per capita expenditure on police by state and local government
5. Ex1 : 1959 per capita expenditure on police by state and local government
6. LF : Labor force participation rate per 1000 civilian urban males age 14-24
7. M : The number of males per 1000 females
8. N : State population size in hundred thousands
9. NW : The number of non-whites per 1000 population
10. U1 : Unemployment rate of urban males per 1000 of age 14-24
11. U2 : Unemployment rate of urban males per 1000 of age 35-39
12. W : Median value of transferable goods and assets or family income in tens of $
13. X : The number of families per 1000 earning below 1/2 the median income

Un exemple : USCrime (suite)

R	Age	S	Ed	Ex0	Ex1	LF	M	N	NW	U1	U2	W	X
79.1	151	1	91	58	56	510	950	33	301	108	41	394	261
163.5	143	0	113	103	95	583	1012	13	102	96	36	557	194
57.8	142	1	89	45	44	533	969	18	219	94	33	318	250
196.9	136	0	121	149	141	577	994	157	80	102	39	673	167
123.4	141	0	121	109	101	591	985	18	30	91	20	578	174
68.2	121	0	110	118	115	547	964	25	44	84	29	689	126
96.3	127	1	111	82	79	519	982	4	139	97	38	620	168
155.5	131	1	109	115	109	542	969	50	179	79	35	472	206
85.6	157	1	90	65	62	553	955	39	286	81	28	421	239
70.5	140	0	118	71	68	632	1029	7	15	100	24	526	174
167.4	124	0	105	121	116	580	966	101	106	77	35	657	170
84.9	134	0	108	75	71	595	972	47	59	83	31	580	172
51.1	128	0	113	67	60	624	972	28	10	77	25	507	206
66.4	135	0	117	62	61	595	986	22	46	77	27	529	190
79.8	152	1	87	57	53	530	986	30	72	92	43	405	264
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Expliquer la variable R par les autres attributs.

Modélisation de la régression

- Une variable aléatoire $Z = (X, Y)$ à valeurs dans $\mathbb{R}^n \times \mathbb{R}$
- Les exemples sont des couples $(x, y) \in \mathbb{R}^n \times \mathbb{R}$ tirés selon la distribution jointe
  $P(Z = (x, y)) = P(X = x)P(Y = y|X = x)$.
- Un échantillon S est un ensemble fini d'exemples $\{(x_1, y_1), \ldots, (x_l, y_l)\}$ i.i.d. selon P.

Modélisation de la régression (suite)

On cherche une fonction : $f : \mathbb{R}^n \to \mathbb{R}$.
Fonction de perte (loss function)

$$L(y, f(x)) = (y - f(x))^2.$$

La fonction risque (ou erreur) : espérance mathématique de la fonction de perte.

$$R(f) = \int L(y, f(x)) dP(x, y) = \int_{\mathbb{R}^n \times \mathbb{R}} (y - f(x))^2 dP(x, y).$$

Le problème général de la régression :

étant donné un échantillon $S = \{(x_1, y_1), \ldots, (x_l, y_l)\}$,
trouver un classifieur f qui minimise le risque $R(f)$.

La fonction de régression

- Il existe une fonction qui minimise l'écart quadratique moyen : la *fonction de régression*

$$r(x) = \int_Y y dP(y|x)$$

- pour chaque x, $r(x)$ est égal à la moyenne des observations
- Comme la fonction de Bayes en classification, la fonction de régression est le plus souvent inaccessible.

Minimisation du risque empirique

- Le risque empirique $R_{emp}(f)$ de f est la moyenne des carrés des écarts à la moyenne de f calculée sur S :

$$R_{emp}(f) = \frac{1}{l}\sum_{i=1}^{l}(y_i - f(x_i))^2.$$

- Principe de minimisation du risque empirique : calculer

$$\underset{f}{ArgMin} R_{emp}(f)$$

Méthode des moindres carrés.

Régression linéaire

On suppose que

$$Y = \langle \alpha, X \rangle + \beta + \epsilon$$

où

- X prend ses valeurs dans $\mathbb{R}^n$,
- $\alpha \in \mathbb{R}^n$ et $\beta \in \mathbb{R}$,
- ϵ est une variable aléatoire telle que $E(\epsilon) = 0$ et $V(\epsilon) = \sigma^2$ (variance indépendante de X).

La fonction de régression est

$$r(x) = \langle \alpha, x \rangle + \beta = \alpha_1 x_1 + \ldots \alpha_n x_n + \beta.$$

Régression linéaire - cas $n = 1$

On suppose que

- X prend des valeurs dans $\mathbb{R}$,
- $Y = \alpha X + \beta + \epsilon$ où $E(\epsilon) = 0$ et $V(\epsilon) = \sigma^2$ (variance indépendante de X).

La fonction de régression est

$$r(x) = \alpha x + \beta.$$

Régression linéaire : estimateurs des moindres carrés

Soit $S = \{(x_1, y_1), \ldots, (x_l, y_l)\}$ un échantillon. Les valeurs de $\hat{\alpha}$ et $\hat{\beta}$ qui minimisent

$$\sum_{i=1}^{l} (y_i - (\hat{\alpha} x_i + \hat{\beta}))^2$$

sont

$$\hat{\alpha} = \frac{\sum_{i=1}^{l} (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{l} (x_i - \overline{x})^2} \text{ et } \hat{\beta} = \overline{y} - \hat{\alpha}\overline{x}$$

où

$$\overline{x} = \frac{1}{l} \sum_{i=1}^{l} x_i \text{ et } \overline{y} = \frac{1}{l} \sum_{i=1}^{l} y_i.$$

Régression linéaire : estimateurs des moindres carrés (suite)

La fonction de régression estimée est alors

$$\hat{r}(x) = \hat{\alpha}x + \hat{\beta}.$$

Les erreurs estimées sont

$$\widehat{\epsilon_i} = y_i - \widehat{y_i} = y_i - (\hat{\alpha}x_i + \hat{\beta}).$$

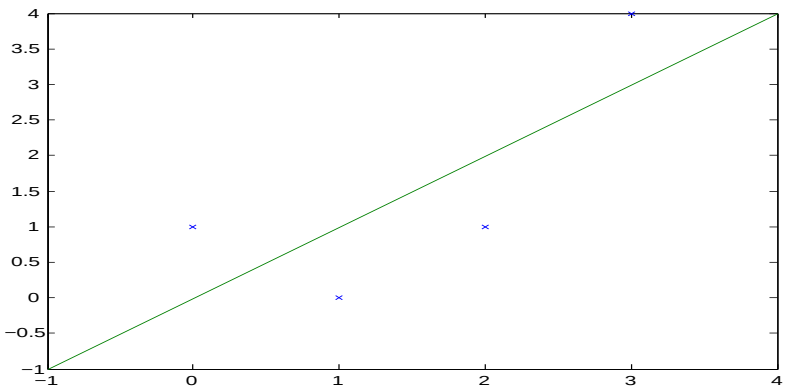
La variance estimée est

$$\hat{\sigma}^2 = \frac{1}{l-2}\sum_{i=1}^{l} \widehat{\epsilon_i}^2.$$

Estimateurs des moindres carrés : exemple

Soit $S = \{(0,1),(1,0),(2,1),(3,4))\}$ un échantillon.
On trouve

$$\overline{x} = 3/2, \overline{y} = 3/2, \hat{\alpha} = 1 \text{ et } \hat{\beta} = 0.$$



On a $\widehat{\epsilon_1} = 1, \widehat{\epsilon_2} = -1, \widehat{\epsilon_3} = -1, \widehat{\epsilon_4} = 1$ et $\hat{\sigma}^2 = 2$.

Propriétés de l'estimateur des moindres carrés

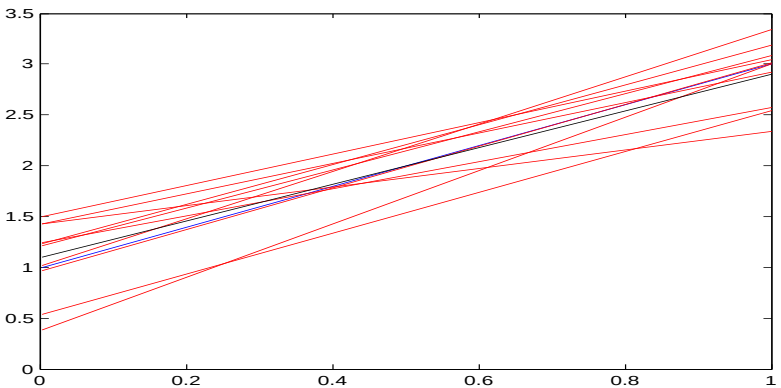
- $\hat{\alpha}, \hat{\beta}$ et $\hat{\sigma}^2$ sont des *estimateurs non biaisés* de α, β et σ^2 : si l'on répète un grand nombre d'expériences avec le même modèle, les moyennes des estimations convergent vers les paramètres du modèle.
- $\hat{\alpha}, \hat{\beta}$ et $\hat{\sigma}^2$ sont des *estimateurs consistants* de α, β et σ^2 : plus on dispose d'observations, plus les estimations se rapprochent des paramètres du modèle.
- si ϵ suit une loi normale, l'estimateur des moindres carrés est aussi l'*estimateur du maximum de vraisemblance* : celui qui maximise la probabilité des observations.

Estimateur non biaisé : illustration

X prend 11 valeurs équidistantes dans $[0, 1]$;
$Y = 2 * X + 1 + Norm(0, 1)$.

On réalise 10 expériences.

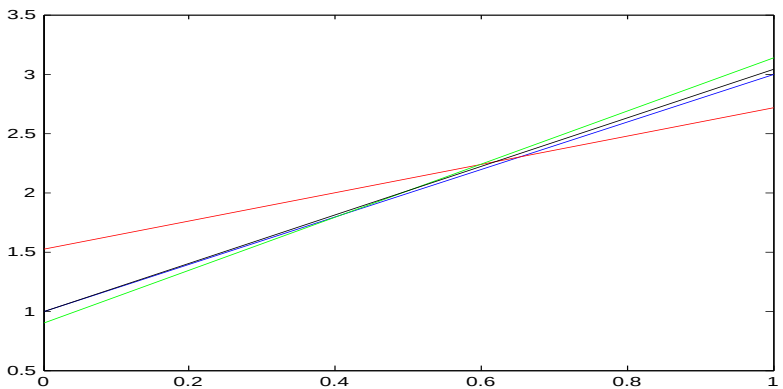
- en bleu : la droite de régression
- en rouge : chaque estimation
- en noir : la moyenne des estimations.



Estimateur consistant : illustration

X prend N valeurs équidistantes dans $[0, 1]$;
$Y = 2 * X + 1 + Norm(0, 1)$.

- en bleu : la droite de régression
- en rouge : $N = 11$
- en vert : $N = 101$
- en noir : $N = 1001$.



Prédiction

Soit $\hat{r}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$ la fonction de régression estimée.
Si l'on observe la nouvelle valeurs x^*, on prédira la valeur $\hat{y}^* = \hat{r}(x^*)$.
Soit

$$\hat{\xi_n}^2 = \hat{\sigma}^2 \left(\frac{\sum_{i=1}^n (x_i - x^*)^2}{n \sum_{i=1}^n (x_i - \overline{x})^2} + 1 \right).$$

Un intervalle de confiance approché au niveau $1 - \alpha$ pour y^* est

$$[\hat{y}^* - u_{\alpha/2}\hat{\xi_n}, \hat{y}^* + u_{\alpha/2}\hat{\xi_n}]$$

Régression linéaire multivariée

Soit $\{(x_1, y_1), \ldots, (x_l, y_l)\} \subset \mathbb{R}^{n+1}$ l'échantillon d'apprentissage.

Soit X la matrice $l \times (n+1)$ dont la i-ème ligne est $x_i, 1$.
Soit Y le vecteur colonne composé des étiquettes y_i.

L'estimateur des moindres carrés est

$$\begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = (X^T X)^{-1} X^T Y$$

où X^T désigne la matrice transposée de X.

Si $X^T X$ n'est pas inversible, ou si $det(X^T X) \simeq 0$, ... il est nécessaire de transformer le problème.

Exemple :

$S = \{((0,0),-1),((0,1),1),((1,0),1),((1,1),1)\}$.
On a

$$X = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, X^T = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \text{ et } Y = \begin{pmatrix} -1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

On vérifie que

$$X^T X = \begin{pmatrix} 2 & 1 & 2 \\ 1 & 2 & 2 \\ 2 & 2 & 4 \end{pmatrix}, (X^T X)^{-1} = \begin{pmatrix} 1 & 0 & -1/2 \\ 0 & 1 & -1/2 \\ -1/2 & -1/2 & 3/4 \end{pmatrix}$$

$$(X^T X)^{-1} X^T Y = \begin{pmatrix} 1 \\ 1 \\ -1/2 \end{pmatrix}$$

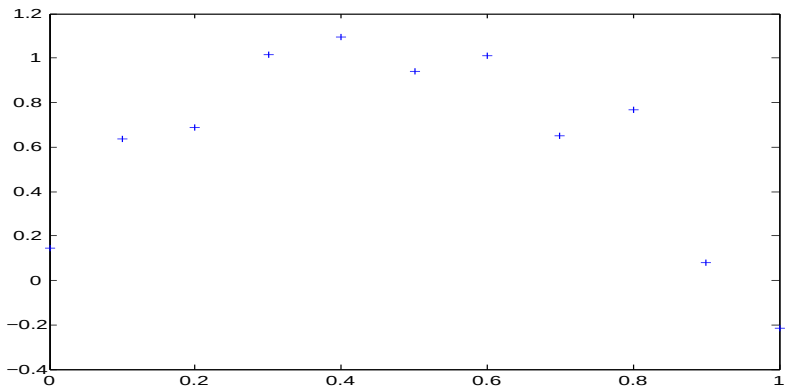
soit

$$\hat{\alpha} = (1,1) \text{ et } \hat{\beta} = -1/2.$$

Un exemple

On observe les données suivantes :

0	0.1000	0.2000	0.3000	0.4000	0.5000	0.6000	0.7000	0.8000	0.9000	1.0000
0.1434	0.6351	0.6856	1.0160	1.0964	0.9393	1.0098	0.6516	0.7655	0.0796	-0.2138



Les données ne semblent pas alignées : on les transforme par les fonctions

$$h_1(x) = x, h_2(x) = x^2 \text{ et } h_3(x) = x^3.$$

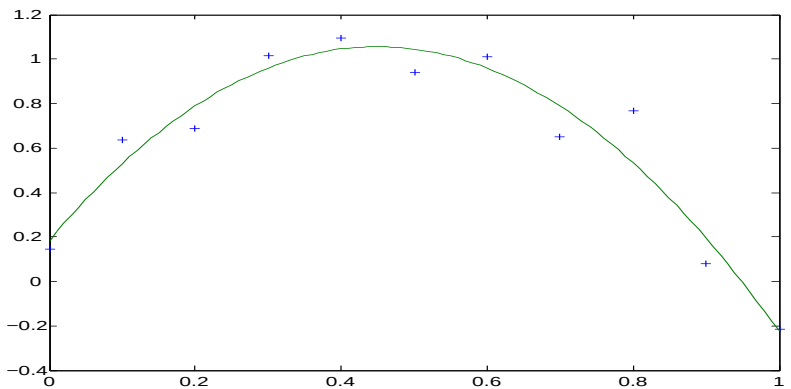
Un exemple (suite)

Une régression linéaire multivariée permet de trouver

$$\beta = 0.1848, \alpha_1 = 3.8960, \alpha_2 = -4.3942 \text{ et } \alpha_3 = 0.0878$$

soit le polynôme

$$p(x) = 0.1848 + 3.8960x - 4.3942x^2 + 0.0878x^3.$$



Ridge regression

Autre idée : pénaliser la taille du modèle. On cherche à minimiser

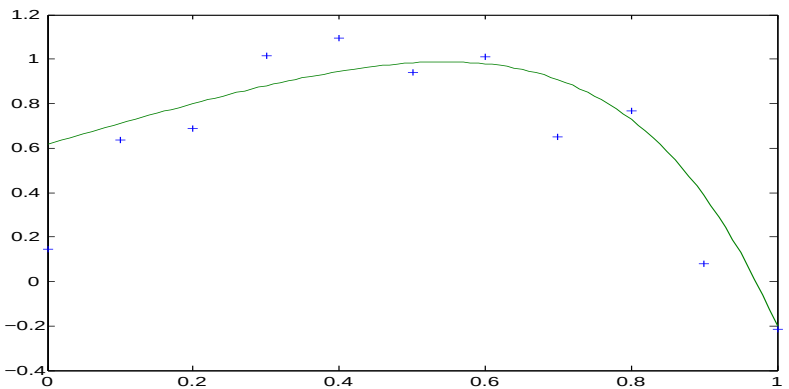
$$\sum_{i=1}^{l}(y_i - (\alpha x_i + \beta))^2 + C(\beta^2 + ||\alpha||^2).$$

- Se résoud aussi simplement que la régression multivariée sans pénalisation.
- Mais comment trouver la valeur de C ? Par exemple, par validation croisée.

Un exemple (suite)

Avec les monômes $h_i(x) = x^i$ pour $1 \leq i \leq 10$ et $C = 0.1$, on trouve les coefficients

$$0.6189\ ;\ 0.9368\ ;\ -0.0655\ ;\ -0.3620\ ;\ -0.4026$$
$$-0.3529\ ;\ -0.2745\ ;\ -0.1906\ ;\ -0.1095\ ;\ -0.0341\ ;\ 0.0350.$$



Lasso

Autre idée : on cherche à minimiser

$$\sum_{i=1}^{l}(y_i - (\hat{\alpha}x_i + \hat{\beta}))^2 + C(|\beta| + \sum |\alpha_i|).$$

- Plus difficile à résoudre.
- Un grand nombre de coefficients s'annule.
- Il faut toujours déterminer une bonne valeur pour C.

Learning with Reproducing Kernels

Reference : B. Schölkopf, A. Smola (2002).
Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.

Regression - Functional Learning

$$y_i = f(x_i) + \epsilon_i$$

- Supervised learning

  $\longrightarrow$ Data : N training examples $\{(\mathbf{x_1}, \mathbf{y_1}), \ldots, (\mathbf{x_N}, \mathbf{y_N})\}$
  $\longrightarrow$ Goal : learn f

Predictor	$\longmapsto$	Response	Model
$\mathbb{R}^d$		$\{-1, 1\}$	Binary Classification
$\mathbb{R}^d$		$\{1, 2, 3, \ldots\}$	Multiclass Classification
$\mathbb{R}^d$		$\mathbb{R}$	Multiple Regression

- Linear model : $Y = AX + B$ - parametric -
- Nonlinear/**Nonparametric** estimation

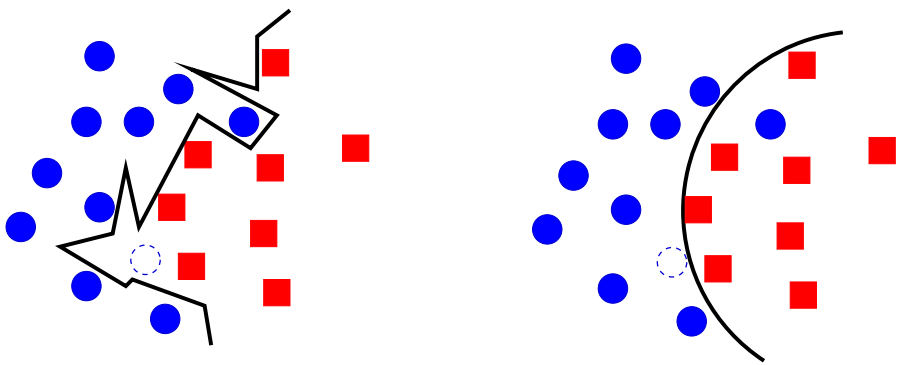
Regression - Functional Learning

- Minimization problem

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

- Overfitting problem : Performance de généralisation

  $\longrightarrow$ compromis adéquation aux données et complexité



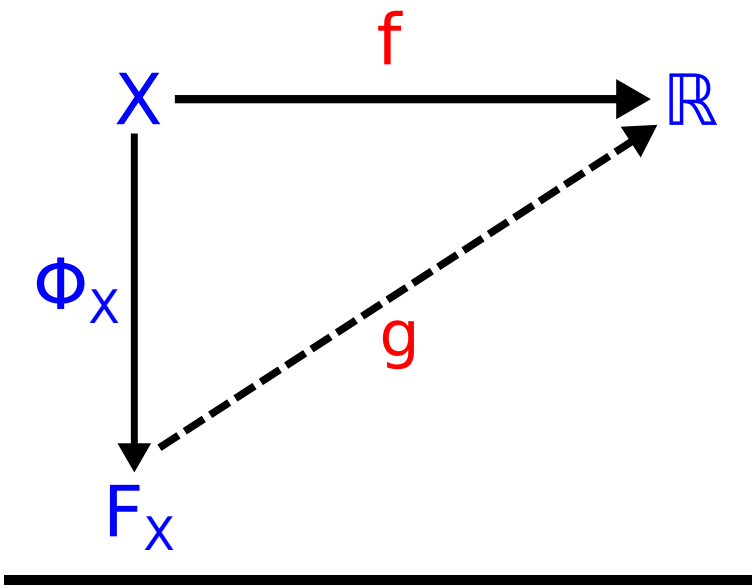
- Regularized minimization

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{F}}^2$$

Regression - Kernel-based Learning

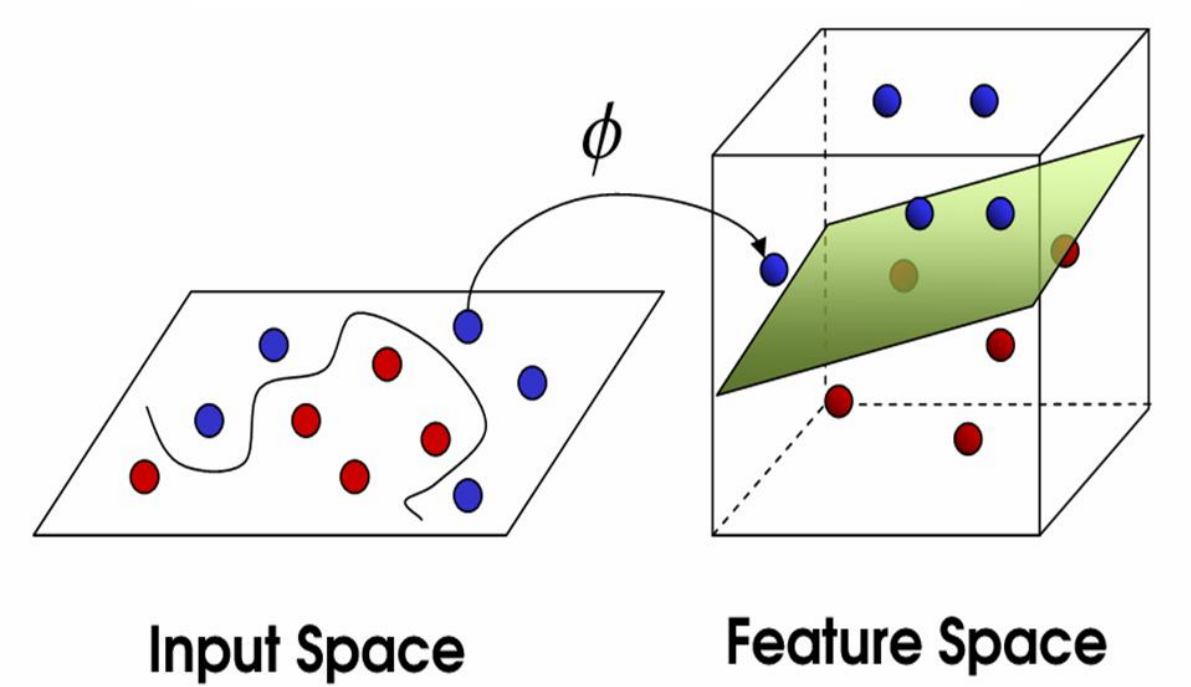
- $y_i = f(x_i) + \epsilon_i$

Scalar-valued



- **RKHS associated with a positive definite kernel k gives a desired feature space ! !**

RKHS



RKHS associated with a positive definite kernel k gives a desired feature space ! !

- Kernel-based Learning

- Kernel : function $k : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$
- Gram matrix : $K = (k_{ij})_{1 \leq i,j \leq n}$, $k_{ij} = k(x_i, x_j)$
- Positive kernel : $\forall c_1, \ldots, c_m \in \mathbb{R}$, $\sum_{i,j} c_i c_j k(x_i, x_j) \geq 0$
- Kernel trick : $k(x, y) = \langle \Phi(x), \Phi(y) \rangle$
- Reproducing property : $f(x) = \langle f, k(x, \cdot) \rangle$
- Representer theorem : $f(\cdot) = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot)$, $\alpha_i \in \mathbb{R}$
- Solution (least sqaures case) : $(K + \lambda I)\alpha = Y$

Reproducing kernel

$(x_i, y_i)_{i=1}^n \in \mathcal{G}_x \times \mathcal{G}_y$, where $\mathcal{G}_y = \mathbb{R}$

Definition

A scalar-valued kernel $K_{\mathcal{F}}(w, z)$ on $\mathcal{G}_x$ is a function $K_{\mathcal{F}}(.,.) : \mathcal{G}_x \times \mathcal{G}_x \longrightarrow \mathbb{R}$;

- *$K_{\mathcal{F}}$ is symmetric if $K_{\mathcal{F}}(w, z) = K_{\mathcal{F}}(z, w)$,*
- *it is positive-definite on $\mathcal{G}_x$ if for any $\{(w_i, u_i)_{i=1,\ldots,r}\} \in \mathcal{G}_x \times \mathcal{G}_y$*

$$\sum_{i,j} u_i u_j K_{\mathcal{F}}(w_i, w_j) \geq 0$$

RKHS (1/4)

Definition

A Hilbert space $\mathcal{F}$ of functions from $\mathcal{G}_x$ to $\mathbb{R}$ is called a reproducing kernel Hilbert space if there is a positive-definite kernel $K_{\mathcal{F}}(w,z)$ on $\mathcal{G}_x^2$ such that :

- *the function $z \longmapsto K_{\mathcal{F}}(w,z)$ belongs to $\mathcal{F}$ for every choice of $w \in \mathcal{G}_x$*
- *for every $f \in \mathcal{F}, \langle f, K_{\mathcal{F}}(w,.)\rangle_{\mathcal{F}} = f(w)$*

(reproducing property)

RKHS (2/4)

Theorem

If a Hilbert space $\mathcal{F}$ of functions on $\mathbb{R}$ admits a reproducing kernel, then the reproducing kernel $K_{\mathcal{F}}(w, z)$ is uniquely determined by the Hilbert space $\mathcal{F}$.

Proof :

$$\langle K'_{\mathcal{F}}(w', .), K_{\mathcal{F}}(w, .)\rangle_{\mathcal{F}} = ... \quad (1)$$

$$\langle K'_{\mathcal{F}}(w', .), K_{\mathcal{F}}(w, .)\rangle_{\mathcal{F}} = ...$$

$$= ... \quad (2)$$

(1) and $(2) \Longrightarrow K_{\mathcal{F}}(w, z) \equiv K'_{\mathcal{F}}(w, z)$ $\square$

RKHS (3/4)

Theorem

A kernel $K_{\mathcal{F}}(w, z)$ on $\mathcal{G}_x^2$ is the reproducing kernel of some Hilbert space $\mathcal{F}$, if and only if it is positive definite.

Proof : RKHS $\Rightarrow$ positive definite kernel

$$\begin{aligned}
&\sum_{i,j=1}^{n} u_i u_j K_{\mathcal{F}}(w_i, w_j) \\
&= \ldots \\
&= \ldots \\
&= \ldots \geq 0 \qquad\qquad \square
\end{aligned}$$

RKHS (4/4)

Proof : RKHS $\Leftarrow$ positive definite kernel

- $\mathcal{F}_0,\ \forall f \in \mathcal{F}_0,\ f(.) = \sum_{i=1}^{n} K_{\mathcal{F}}(w_i, .)\alpha_i$
- $\langle f(.), g(.) \rangle_{\mathcal{F}_0} = \sum_{i,j=1}^{n} \alpha_i \beta_j K_{\mathcal{F}}(w_i, z_j)$
- $(\mathcal{F}_0, \langle ., . \rangle_{\mathcal{F}_0})$ is a pre-Hilbert space
- $\mathcal{F}$ the completion of $\mathcal{F}_0$ space via Cauchy sequences

Reproducing kernels - Examples

- Examples of positive definite kernels

$\longrightarrow$ Polynomial kernel : $k(x,y) = (1 + \sum_{i=1}^{n} x_i y_i)^d$

$\longrightarrow$ Exponential kernel : $k(x,y) = \exp(\beta x^\top y)$
$= 1 + \beta x^\top y + \frac{\beta^2}{2!}(x^\top y)^2 + \ldots$

$\longrightarrow$ Gaussien kernel : $k(x,y) = exp\left(-\frac{||x-y||^2}{2\sigma^2}\right)$

Reproducing kernels - Positive-definitness

- operations that preserve positive definiteness

  $\longrightarrow$ positive combination : $ak_1 + bk_2, \quad (a, b) \geq 0$

  $\longrightarrow$ product : $k_1 k_2, \quad (k_1(x, y)k_2(x, y))$

  $\longrightarrow$ feature map : $k(x, y) = \langle \Phi(x), \Phi(y) \rangle$

  $\longrightarrow$ function combination : $\tilde{k}(x, y) = f(x)k(x, y)f(y)$

- Proofs ?

Kernel Ridge Regression

Reference : C. Saunders, A. Gammerman, V. Vovk (1998).
Ridge regression learning algorithm in dual variable. ICML'98.

Regression with Kernels-Representer theorem (1/2)

Theorem

The solution of the minimization problem

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{F}}^2$$

is achieved by a function of the form

$$f^*(.) = \sum_{i=1}^{n} \alpha_i K_{\mathcal{F}}(x_i, .)$$

Regression with Kernels-Representer theorem (2/2)

Proof :

$$D_h J_\lambda(f) = \lim_{\tau \longrightarrow 0} \frac{J_\lambda(f+\tau h) - J_\lambda(f)}{\tau} = \langle J_\lambda^{'}(f), h \rangle$$

i. $G(f) = \|f\|_{\mathcal{F}}^2$

$$\lim_{\tau \longrightarrow 0} \frac{\|f+\tau h\|_{\mathcal{F}}^2 - \|f\|_{\mathcal{F}}^2}{\tau} = 2\langle f, h \rangle \qquad \Longrightarrow \qquad G^{'}(f) = 2f$$

ii. $H_i(f) = (y_i - f(x_i))^2$

$$\lim_{\tau \longrightarrow 0} \frac{(y_i - f(x_i) - \tau h(x_i))^2 - (y_i - f(x_i))^2}{\tau}$$

$$= -2(y_i - f(x_i))h(x_i) = -2\langle (y_i - f(x_i))K_{\mathcal{F}}(x_i, .), h \rangle_{\mathcal{F}}$$

$$= -2\langle \alpha_i K_{\mathcal{F}}(x_i, .), h \rangle_{\mathcal{F}} \qquad \Longrightarrow \qquad H_i^{'}(f) = 2\alpha_i K_{\mathcal{F}}(x_i, .)$$

(i), (ii), and $J_\lambda^{'}(f^*) = 0 \Longrightarrow f^*(.) = \frac{1}{\lambda} \sum_{i=1}^{n} K_{\mathcal{F}}(x_i, .)\alpha_i$ □

Regression with Kernels - Solution (1/2)

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{F}}^2$$

using the representer theorem

$$\iff \min_{\alpha_i} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{n} \alpha_j K_{\mathcal{F}}(x_i, x_j))^2 + \lambda \| \sum_{j=1}^{n} \alpha_j K_{\mathcal{F}}(., x_j) \|_{\mathcal{F}}^2$$

using the reproducing property

$$\iff \min_{\alpha_i} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{n} \alpha_j K_{\mathcal{F}}(x_i, x_j))^2 + \lambda \sum_{i,j}^{n} \alpha_i \alpha_j K_{\mathcal{F}}(x_i, x_j)$$

Regression with Kernels - Solution (2/2)

Theorem

The the solution of the optimization problem $\min_{\alpha_i} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{n} \alpha_j K_{\mathcal{F}}(x_i, x_j))^2 + \lambda \sum_{i,j}^{n} \alpha_i \alpha_j K_{\mathcal{F}}(x_i, x_j)$ *has the following form :*

$$\alpha = (K_{\mathcal{F}} + \lambda I)^{-1} Y$$

where

- $K_{\mathcal{F}} = (K_{\mathcal{F}}(x_i, xj))_{1 \leq i,j \leq n} \in \mathbb{R}^{n \times n}$ *: kernel matrix*
- $\alpha = (\alpha_i)_{1 \leq i \leq n} \in \mathbb{R}^n$
- $Y = (y_i)_{1 \leq i \leq n} \in \mathbb{R}^n$

PART II

From scalar to complex outputs

Operator-valued Kernels

Intro. - Complex outputs

- **Complex** output = a target variable which is **not** a **scalar**
- **3** categories
  1. Euclidean : (vectors/**functions**) curves, images, ...
  2. Mildly non-Euclidean : points on a manifold and shapes
  3. Strongly non-Euclidean : (**structured** objects) tree, graph, **text**
- Examples
  1. **Multi-task learning**
  2. Multivariate or **functional regression**
  3. Multiple Kernel Learning / Multi-view Learning
  4. **Structured output prediction**

Complex outputs - Learning problems

$$y_i = f(x_i) + \epsilon_i$$

Predictor	$\longmapsto$	Response	Model
$\mathbb{R}^d$		$\{-1, 1\}$	Binary Classification
$\mathbb{R}^d$		$\{1, 2, 3, \ldots\}$	Multiclass Classification
$\mathbb{R}^d$		$\mathbb{R}$	Multiple Regression
$\mathbb{R}^d$		$\mathbb{R}$	Multiple Regression
$\mathbb{R}^d$		$\mathbb{R}^p$	Multivariate Regression-Multi-task
L^2		L^2	Functional Response Regression
Non-numeric		Non-numeric	Structured Output Prediction

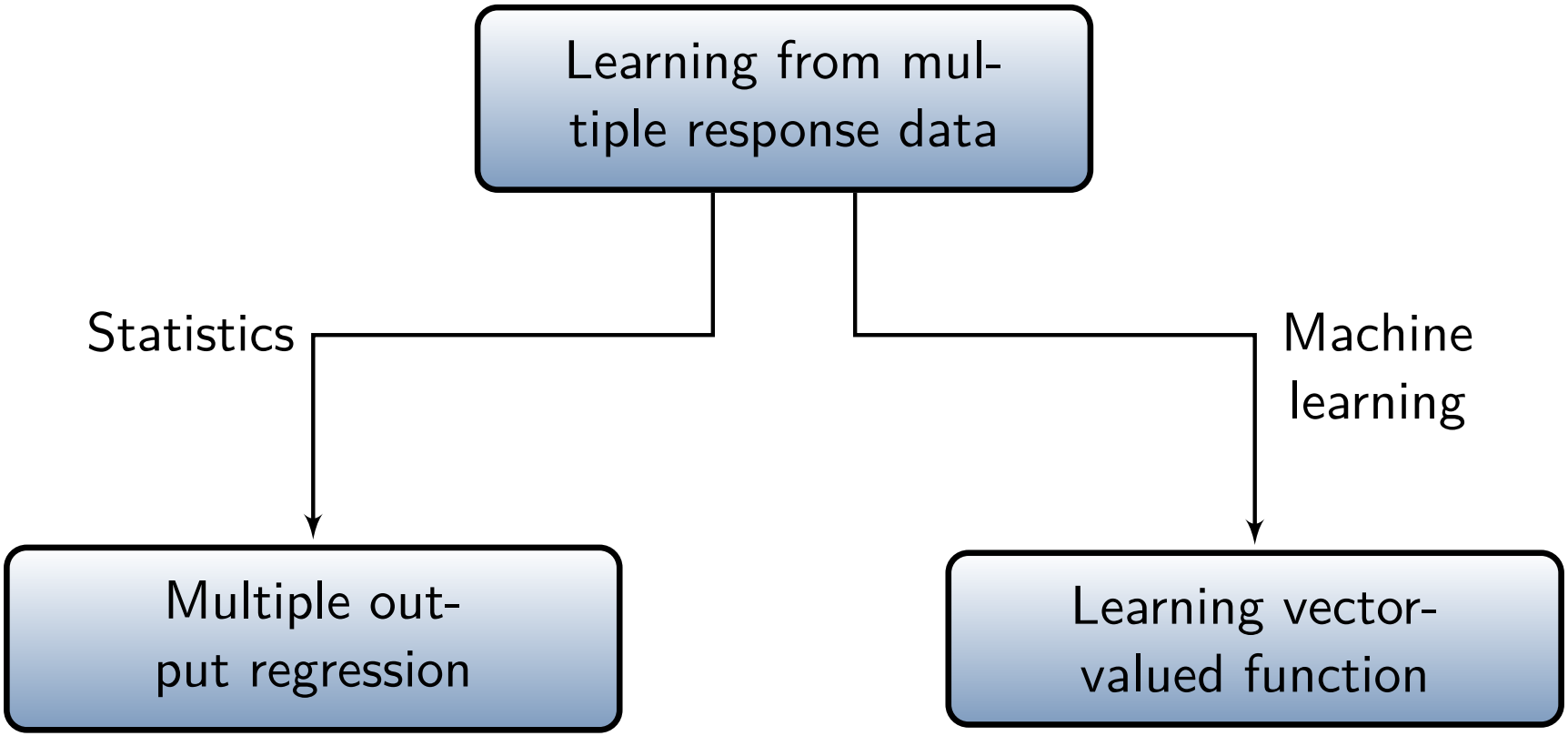
Outline

- **Multi-Task Learning with Kernels**
  - → Matrix-valued Kernels
  - → Vector-valued RKHS
- **Functional Response Regression**
  - → Functional Data Analysis
  - → Operator-valued kernels
  - → Application to Audio Signal Processing
- **Structured Output Prediction**
  - → Kernel Dependency Estimation
  - → Covariance Operator in RKHS
  - → Some Applications

Vector outputs

Reference : C. Micchelli, M. Pontil (2005).
On Learning Vector-valued Functions. Neural Computation, MIT Press.

Learning from vector outputs - Overview



Learning from vector outputs - Matrix learning

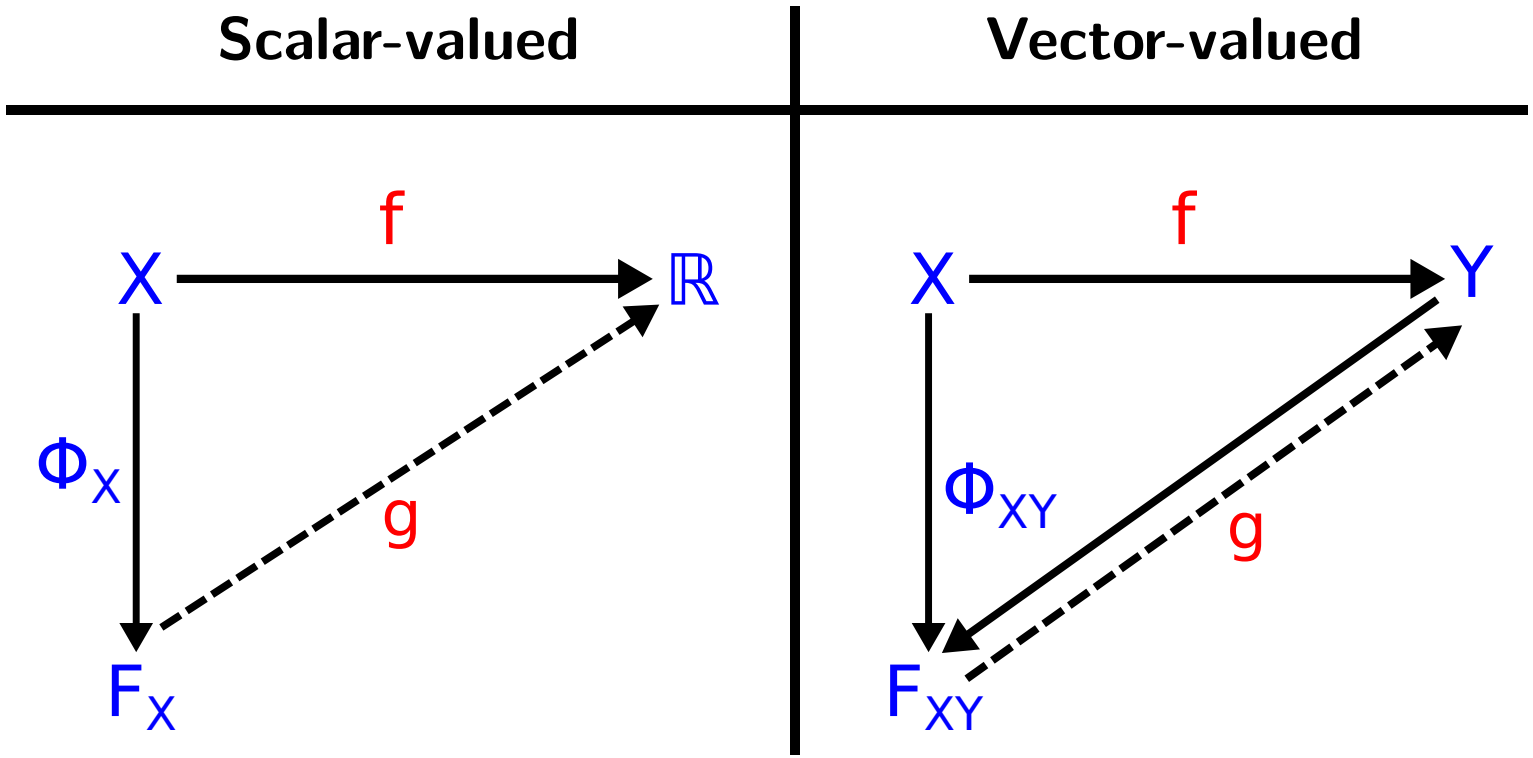
$$y_i = f(x_i) + \epsilon_i$$

Predictor	$\longmapsto$	Response	Model
$\mathbb{R}^d$		$\{-1, 1\}$	Binary Classification
$\mathbb{R}^d$		$\{1, 2, 3, \ldots\}$	Multiclass Classification
$\mathbb{R}^d$		$\mathbb{R}$	Multiple Regression
$\mathbb{R}^d$		$\mathbb{R}^p$	Multivariate Regression
			Multi-task Learning

- Matrix estimation

$$\longrightarrow \quad \min_{f \in \mathcal{F}} \sum_{i=1}^{n} \|y_i - f(x_i)\|^2_{\mathbb{R}^p} + \lambda \|f\|^2_{\mathcal{F}}$$

Matrix learning - Kernel-based approach



- **From scalar-valued to matrix-valued kernels**

Matrix-valued kernels - Definition

- $(x_i, y_i)_{i=1}^n \in \mathcal{G}_x \times \mathcal{G}_y$
- $\mathcal{G}_x = \mathbb{R}^d \quad ; \quad \mathcal{G}_y = \mathbb{R}^p$
- $f \in \mathbb{R}^{d \times p} \quad ; \quad \mathcal{L}(\mathcal{G}_y) = \mathbb{R}^{p \times p}$

Definition

$K_{\mathcal{F}}(.,.) : \mathcal{G}_x \times \mathcal{G}_x \longrightarrow \mathcal{L}(\mathcal{G}_y)$

- $K_{\mathcal{F}}$ *is Hermitian if* $K_{\mathcal{F}}(w, z) = K_{\mathcal{F}}(z, w)^*$,
- *it is nonnegative on* $\mathcal{G}_x$ *if Hermitian and for any* $\{(w_i, u_i)_{i=1,\ldots,r}\} \in \mathcal{G}_x \times \mathcal{G}_y$

$$\sum_{i,j} \langle K_{\mathcal{F}}(w_i, w_j) u_i, u_j \rangle_{\mathcal{G}_y} \geq 0$$

Matrix-valued kernels - Vector-valued RKHS

- Extending real-valued RKHS theory to Multi-task setting (Micchelli & Pontil 2004)
- RKHS of vector-valued functions

Definition

A Hilbert space $\mathcal{F} = \{f : \mathcal{G}_x \longrightarrow \mathcal{G}_y\}$ is called a reproducing kernel Hilbert space if there is an operator-valued kernel $K_{\mathcal{F}}$ such that :

- $h : z \longmapsto K_{\mathcal{F}}(w, z)g \Longrightarrow h \in \mathcal{F}, \ \forall w \in \mathcal{G}_x$ *and* $g \in \mathcal{G}_y$
- $\forall f \in \mathcal{F}, \langle f, K_{\mathcal{F}}(w, .)g\rangle_{\mathcal{F}} = \langle f(w), g\rangle_{\mathcal{G}_y}$ *(reproducing property)*

Operator-valued kernels - Uniqueness & Bijection

Lemma

$\mathcal{F}$ vector-valued RKHS $\Longrightarrow$ *$K_{\mathcal{F}}(w,z)$ is unique*

- *Proof :*
  - $\langle K'(w',.)g', K(w,.)g\rangle_{\mathcal{F}} = \langle K'(w',w)g', g\rangle_{\mathcal{G}_y}$
  - $\langle K(w,.)g, K'(w',.)g'\rangle_{\mathcal{F}} = \langle K(w,w')g, g'\rangle_{\mathcal{G}_y}$
    $= \langle g, K(w,w')^* h\rangle_{\mathcal{G}_y} = \langle g, K(w',w)g'\rangle_{\mathcal{G}_y}$

Theorem

$K_{\mathcal{F}}(w,z)$ *nonnegative* $\Longleftrightarrow$ *RKHS $\mathcal{F}$*

- *Proof :*

$\Leftarrow$ $\displaystyle\sum_{i,j=1}^{n} \langle K(w_i,w_j)u_i, u_j\rangle_{\mathcal{G}_y} = \sum_{i,j=1}^{n} \langle K(w_i,.)u_i, K(w_j,.)u_j\rangle_{\mathcal{F}}$

$\Rightarrow$ $\mathcal{F}_0,\ \forall f \in \mathcal{F}_0,\ f(.) = \displaystyle\sum_{i=1}^{n} K_{\mathcal{F}}(w_i,.)\alpha_i$

Matrix-valued kernels - Construction

- Multi-task kernel $\quad \Longrightarrow \quad K(w,z) = k(w,z)T$
  - k : real-valued kernel
  - $T \in \mathbb{R}^{p \times p}$
  - T : diagonal matrix + low rank matrix (finite dimension)
- Operations that preserve nonnegativeness

Theorem

$K_1(.,.)$ and $K_2(.,.)$ two nonnegative kernels

i. *$K_1 + K_2$ is a nonnegative kernel*

ii. *If $K_1 K_2 = K_2 K_1$ then $K_1 K_2$ is a nonnegative kernel*

iii. *$K(x,y) = T(y) K_1(x,y) T(x)^*$ is a nonnegative kernel*

Optimization problem - Representer theorem

Theorem

The solution of the minimization problem

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \|y_i - f(x_i)\|^2_{\mathcal{G}_y} + \lambda \|f\|^2_{\mathcal{F}}$$

is achieved by a function of the form

$$f^*(.) = \sum_{i=1}^{n} K_{\mathcal{F}}(x_i, .)\beta_i \quad , \quad \beta_i \in \mathbb{R}^p$$

Optimization problem - Solution

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \|y_i - f(x_i)\|^2_{\mathcal{G}_y} + \lambda \|f\|^2_{\mathcal{F}}$$

using the representer theorem & the reproducing property

$$\Longleftrightarrow \quad \min_{\beta_i \in \mathcal{G}_y} \sum_{i=1}^{n} \|y_i - \sum_{j=1}^{n} K_{\mathcal{F}}(x_i, x_j)\beta_j\|^2_{\mathcal{G}_y} + \lambda \sum_{i,j}^{n} \langle K_{\mathcal{F}}(x_i, x_j)\beta_i, \beta_j \rangle_{\mathcal{G}_y}$$

- Analytic solution

$$\rightarrow \ (\mathcal{K} + \lambda I)\beta = y \quad ; \quad \beta \in (\mathcal{G}_y)^n \ \text{ and } \ \mathcal{K} \in [\mathcal{L}(\mathcal{G}_y)]^{n \times n}$$

Optimization problem - Kernel matrix inversion

- $K(x_i, x_j) = G(x_i, x_j)T, \quad \forall x_i, x_j \in \mathcal{G}_x$

- Kronecker product

$$\rightarrow \mathcal{K} = \begin{pmatrix} G(x_1, x_1)T & \dots & G(x_1, x_n)T \\ \vdots & \ddots & \vdots \\ G(x_n, x_1)T & \dots & G(x_n, x_n)T \end{pmatrix} = \mathcal{G} \otimes T$$

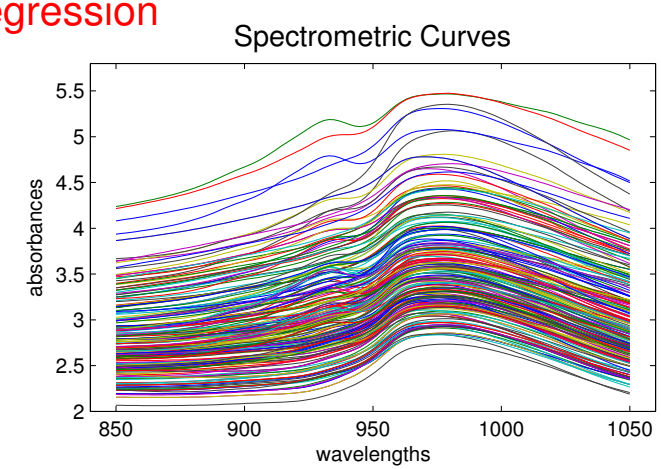
$$\rightarrow \mathcal{K}^{-1} = \mathcal{G}^{-1} \otimes T^{-1}$$

Functional outputs

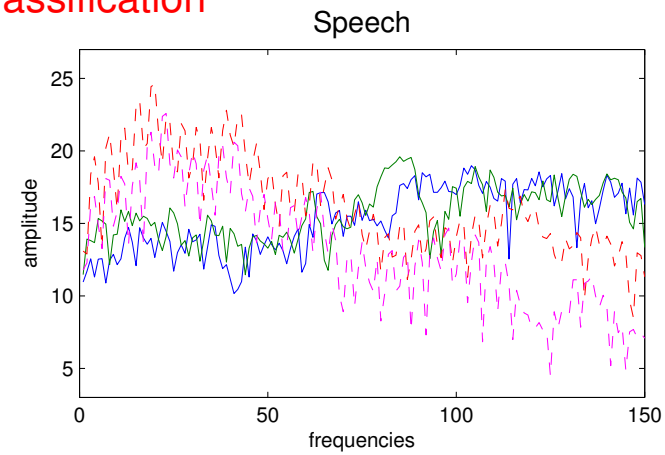
References : J.O. Ramsay, B.W. Silverman (2005).
Functional Data Analysis (2 ed.). New Yok : Springer-Verlag.

Learning on functions - Examples

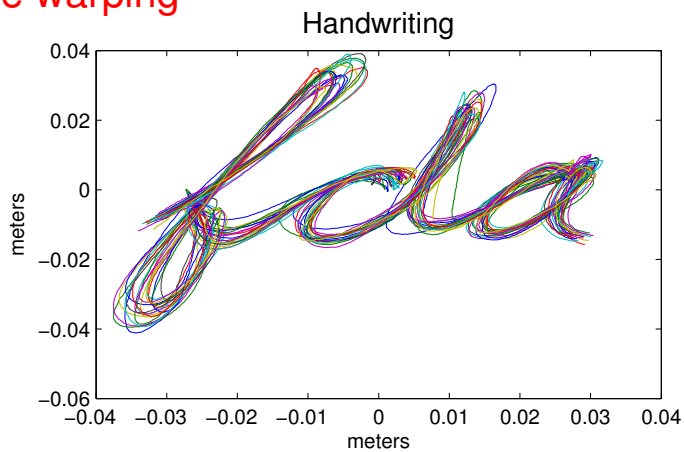
Regression



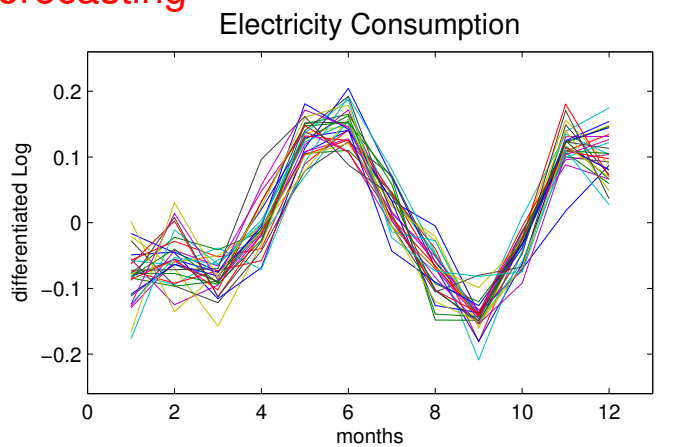
Classification



Time warping



Forecasting



Ramsay and Silverman (2002) - Ferraty and Vieu (2006)

Learning on functions - Operator learning

$$y_i = f(x_i) + \epsilon_i$$

Predictor	$\longmapsto$	Response	Model
$\mathbb{R}^d$		$\mathbb{R}$	Multiple Regression
$\mathbb{R}^d$		$\mathbb{R}^d$	Multivariate Regression
$\mathbf{L}^2$		$\mathbb{R}$	Functional Model Scalar Response
$\mathbb{R}^d$		$\mathbf{L}^2$	Functional Response Model
$\mathbf{L}^2$		$\mathbf{L}^2$	Functional Model Functional Response

- Operator estimation

$$\longrightarrow \quad \min_{f \in \mathcal{F}} \sum_{i=1}^{n} \|y_i - f(x_i)\|^2_{\mathcal{G}_y} + \lambda \|f\|^2_{\mathcal{F}}$$

Operator Learning - From vectors to functions

- Extending real/vector-valued RKHS theory to FDA
- From matrix-valued to operator-valued kernels
- RKHS of operators (or function-valued functions)
- FDA kernel - Nonlinear FDA method
  - $K(w,z) = k(w,z)T \implies T \in \mathcal{L}(\mathcal{G}_y)$ (infinite dimension) ?
  - Concurrent functional linear model
    - $\rightarrow$ $y(t) = \alpha(t) + \beta(t)x(t)$
    - $\rightarrow$ Multiplication operator
  - Functional linear model for functional responses
    - $\rightarrow$ $y(t) = \alpha(t) + \int \beta(s,t)x(s)ds$
    - $\rightarrow$ Hilbert-Schmidt integral operator

Operator-valued kernels - Examples

1. Multiplication operator

$$\begin{array}{rccl} K_{\mathcal{F}}: & \mathcal{G}_x \times \mathcal{G}_x & \longrightarrow & \mathcal{L}(\mathcal{G}_y) \\ & x_1, x_2 & \longmapsto & k_x(x_1, x_2) T^{k_y} \quad ; \quad T_y^h(t) \triangleq h(t) y(t) \end{array}$$

2. Hilbert-Schmidt integral operator

$$\begin{array}{rccl} K_{\mathcal{F}}: & \mathcal{G}_x \times \mathcal{G}_x & \longrightarrow & \mathcal{L}(\mathcal{G}_y) \\ & x_1, x_2 & \longmapsto & k_x(x_1, x_2) T^{k_y} \quad ; \quad T_y^h(t) \triangleq \int h(s,t) y(s) ds \end{array}$$

3. Composition operator

$$\begin{array}{rccl} K_{\mathcal{F}}: & \mathcal{G}_x \times \mathcal{G}_x & \longrightarrow & \mathcal{L}(\mathcal{G}_y) \\ & x_1, x_2 & \longmapsto & C_{\psi(x_1)} C^*_{\psi(x_2)} \quad ; \quad C_\varphi : f \longmapsto f \circ \varphi \end{array}$$

Applications - Speech inversion

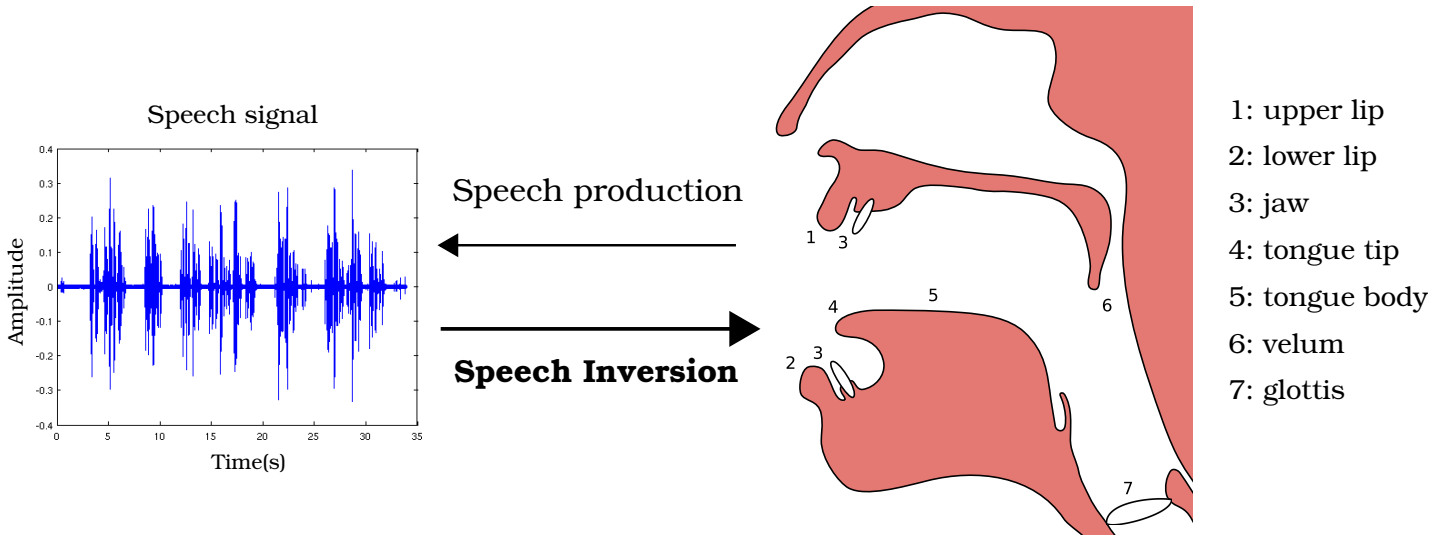
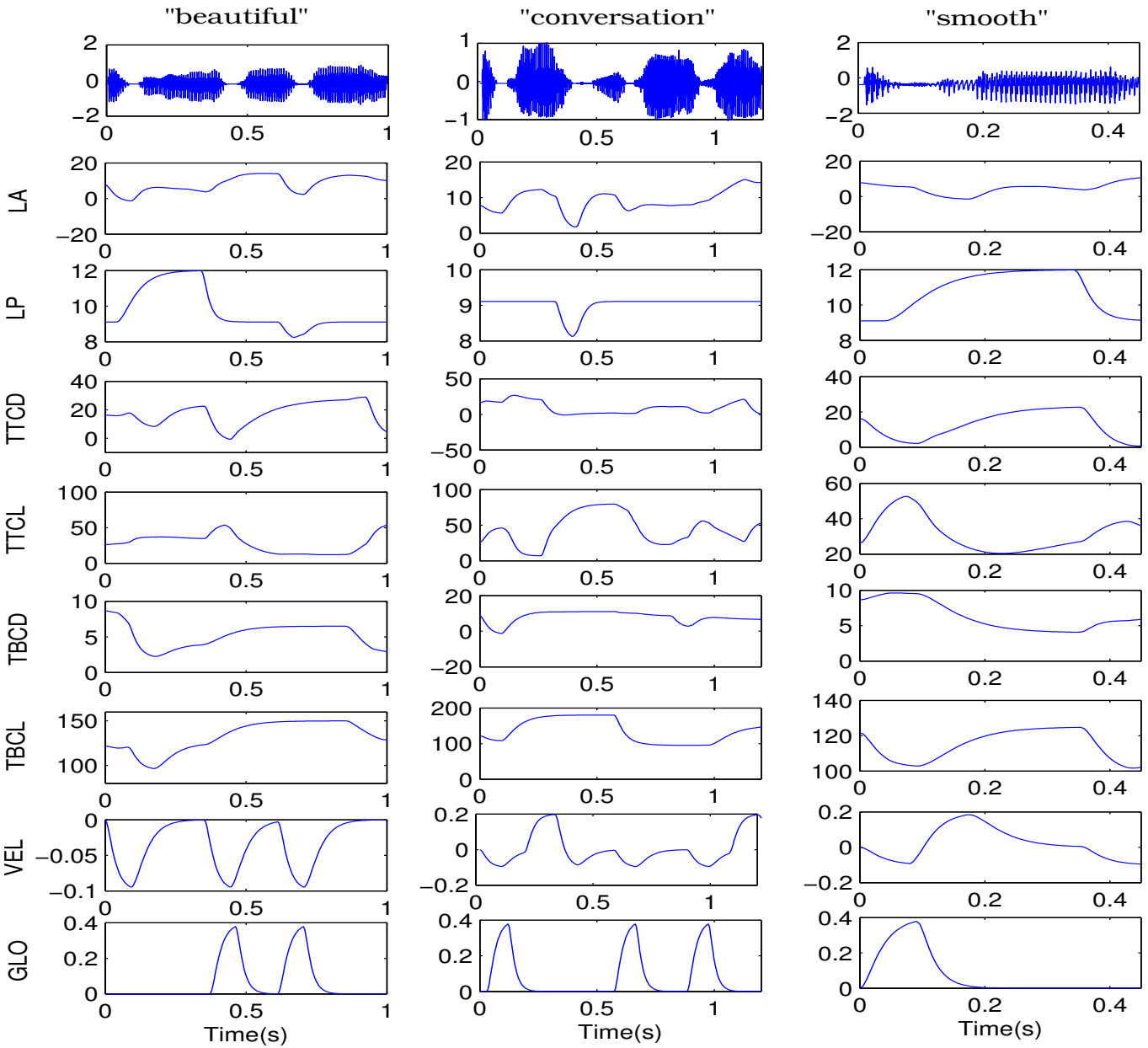


Fig. – Acoustic to articulatory inversion

- speech inversion
  - → learning the acoustic-to-articulatory mapping
  - → from MFCC to Vocal-tract time functions (VTTF)
  - → improving speech technology and understanding
  - → helping individuals with speech and hearing disorders

Applications - Speech inversion



Applications - Speech inversion

Tab. – Average RSSE for the tract variables

VT variables	ε-SVR	Multi-task	functional
LA	2.763	2.341	**1.562**
LP	0.532	**0.512**	0.528
TTCD	3.345	1.975	**1.647**
TTCL	7.752	5.276	**3.463**
TBCD	2.155	2.094	**1.582**
TBCL	15.083	9.763	**7.215**
VEL	0.032	0.034	**0.029**
GLO	**0.041**	0.052	0.064
Total	3.962	2.755	**2.011**

Operator-valued kernels - Feature map

- Operator-valued kernel admits a feature map representation
  - $\rightarrow \langle K(x_1, x_2)y_1, y_2\rangle_{\mathcal{Y}} = \langle \Phi(x_1, y_1), \Phi(x_2, y_2)\rangle_{\mathcal{L}(\mathcal{X},\mathcal{Y})}$
  - $\rightarrow \langle K(x_1, .)y_1, K(x_2, .)y_2\rangle_{\mathcal{F}} = \langle K(x_1, x_2)y_1, y_2\rangle_{\mathcal{Y}}$
- Complex/infinite-dimensional inputs
  - $\rightarrow$ multiple functional data $x_i \in (L^2)^p$
- FDA viewpoint
  - $\rightarrow$ one observation = one continuous curve

Real-valued RKHS

$$\begin{array}{rcl} \Phi_k : (L^2)^p & \rightarrow & \mathcal{L}((L^2)^p, \mathbb{R}) \\ x & \mapsto & k(x, .) \end{array}$$

dim : p $\longrightarrow$ 1

Function-valued RKHS

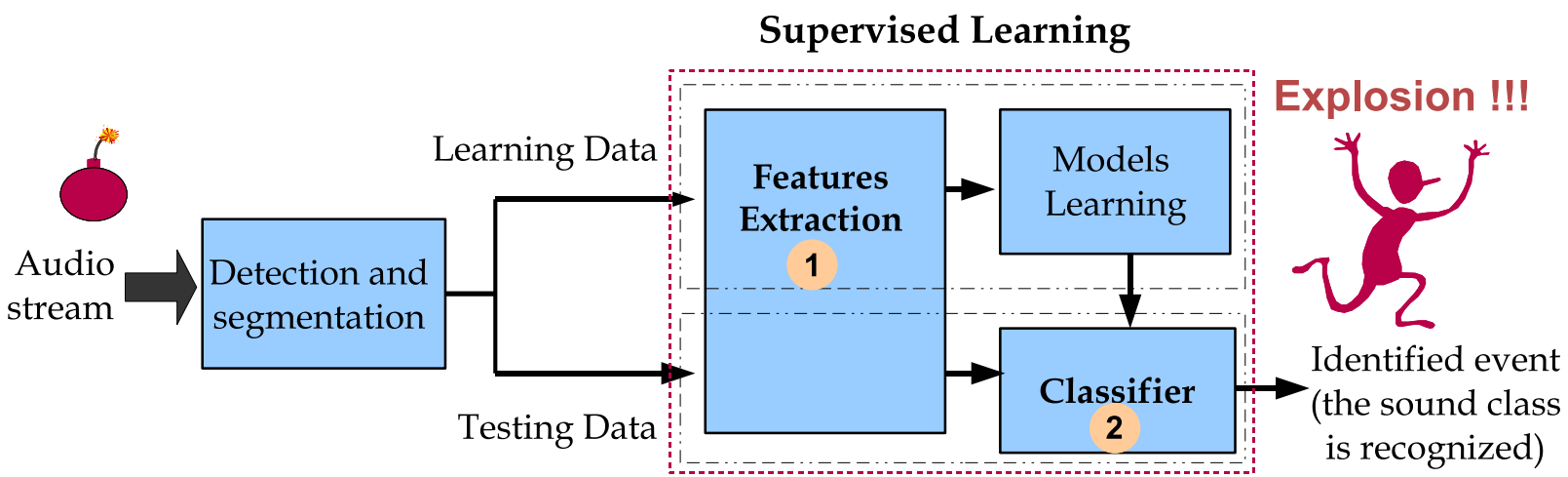
$$\begin{array}{rcl} \Phi_K^y : (L^2)^p & \rightarrow & \mathcal{L}((L^2)^p, L^2) \\ x & \mapsto & K(x, .)y \end{array}$$

dim : p $\longrightarrow$ inf

Applications - Sound recognition

- **Sound Recognition**

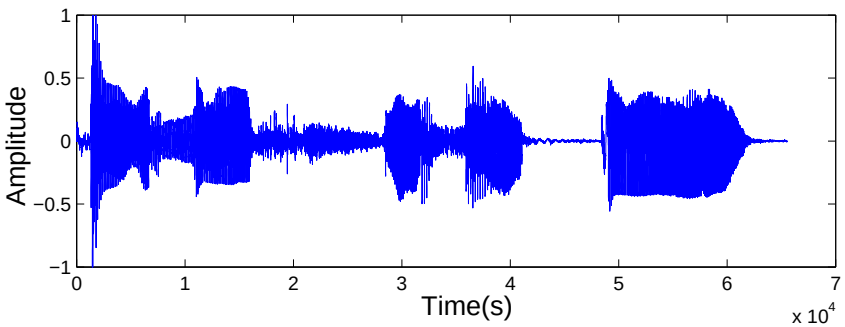
  $\longrightarrow$ Surveillance and security applications

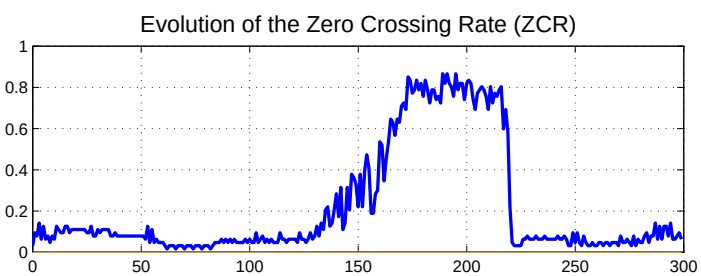


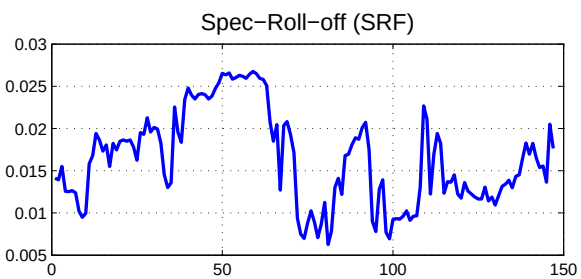
Applications - Sound recognition

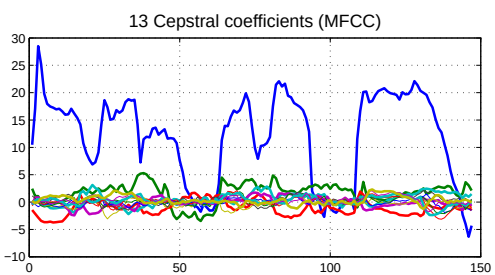
- **Features extraction**

  $\longrightarrow$ temporal, spectral, cepstral, ... characteristics







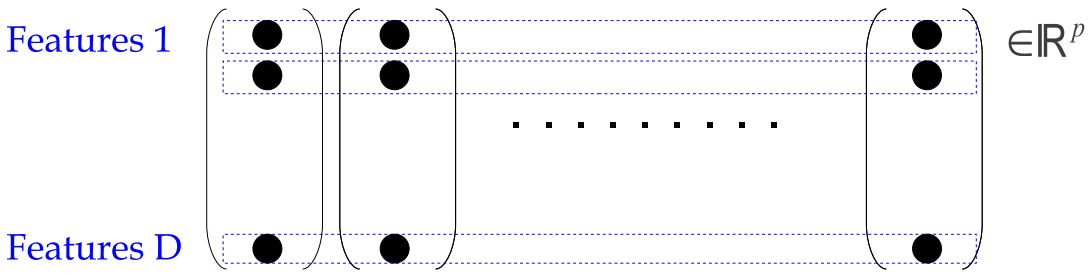


Applications - Sound recognition

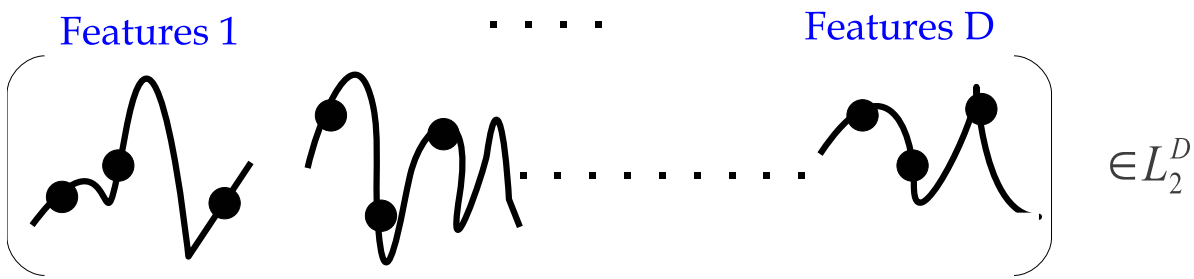
- **Limits** - Multivariate data modeling

  $\longrightarrow$ features contain discrete values of various parameters

  $\longrightarrow$ feature vector $\in \mathbb{R}^{DP}$ by concatenating samples of $\neq$ features



- **Solution** - Multivariate functional data modeling



  $\longrightarrow$ modeling each audio signal by a vector of functions in $(L^2)^D$

Applications - Sound recognition

Tab. – Classes of sounds and number of samples in the database used for performance evaluation.

Classes	Number	Train	Test	Total	Duration (s)
Human screams	C1	40	25	65	167
Gunshots	C2	36	19	55	97
Glass breaking	C3	48	25	73	123
Explosions	C4	41	21	62	180
Door slams	C5	50	25	75	96
Phone rings	C6	34	17	51	107
Children voices	C7	58	29	87	140
Machines	C8	40	20	60	184
Total		327	181	508	18mn 14s

Applications - Sound recognition

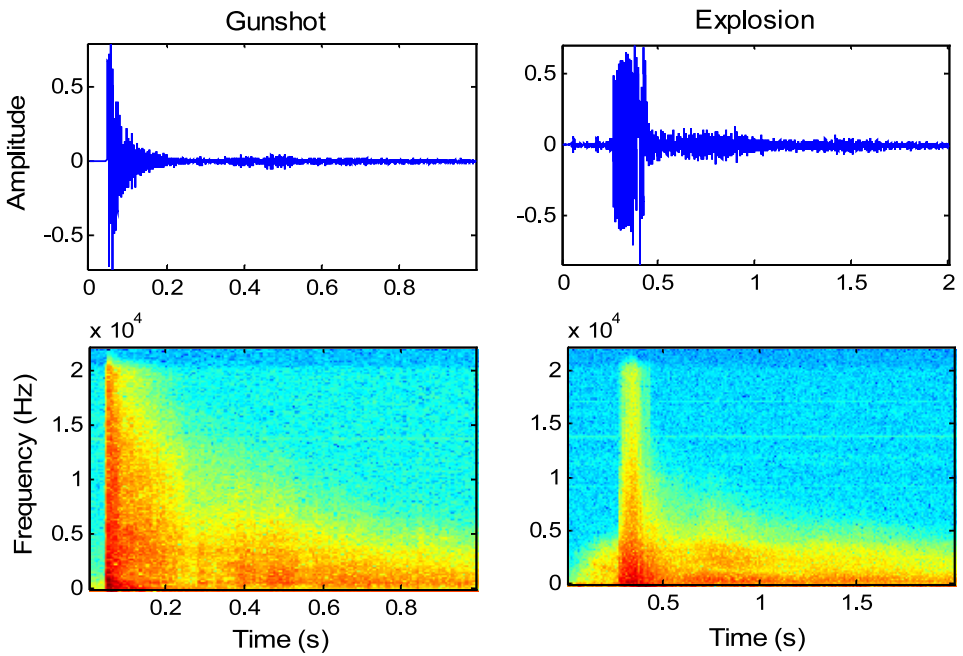


Fig. – Structural similarities between two different classes

Applications - Sound recognition

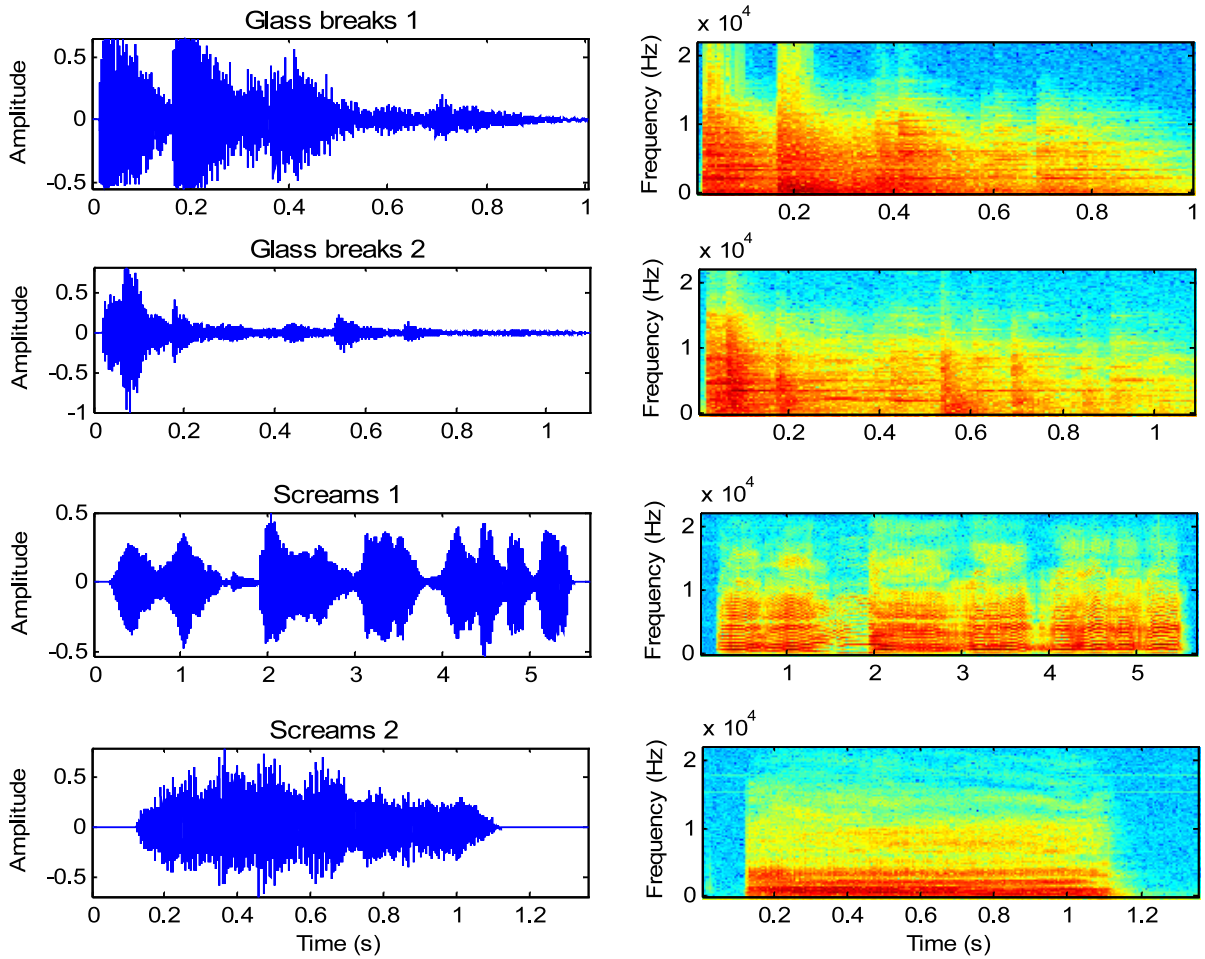


Fig. – Structural diversity inside the same sound class and between classes

Applications - Sound recognition

Tab. – Confusion Matrix obtained when using the Regularized Least Squares Classification (RLSC) algorithm (Rifkin et al, 2003)

	C1	C2	C3	C4	C5	C6	C7	C8
C1	92	4	4.76	0	5.27	11.3	6.89	0
C2	0	52	0	14	0	2.7	0	0
C3	0	20	76.2	0	0	0	17.24	5
C4	0	16	0	66	0	0	0	0
C5	4	8	0	4	84.21	0	6.8	0
C6	4	0	0	0	10.52	86	0	0
C7	0	0	0	8	0	0	69.07	0
C8	0	0	19.04	8	0	0	0	95
Total Recognition Rate = 77.56%								

Applications - Sound recognition

Tab. – Confusion Matrix obtained when using the Functional Regularized Least Squares algorithm

	C1	C2	C3	C4	C5	C6	C7	C8
C1	**100**	0	0	2	0	5.3	3.4	0
C2	0	**82**	0	8	0	0	0	0
C3	0	14	**90.9**	8	0	0	3.4	0
C4	0	4	0	**78**	0	0	0	0
C5	0	0	0	1	**89.47**	0	6.8	0
C6	0	0	0	0	10.53	**94.7**	0	0
C7	0	0	0	0	0	0	**86.4**	0
C8	0	0	9.1	3	0	0	0	**100**
Total Recognition Rate = ***90.18%***								

Other Applications - Meteorology

- Temperature-precipitation data

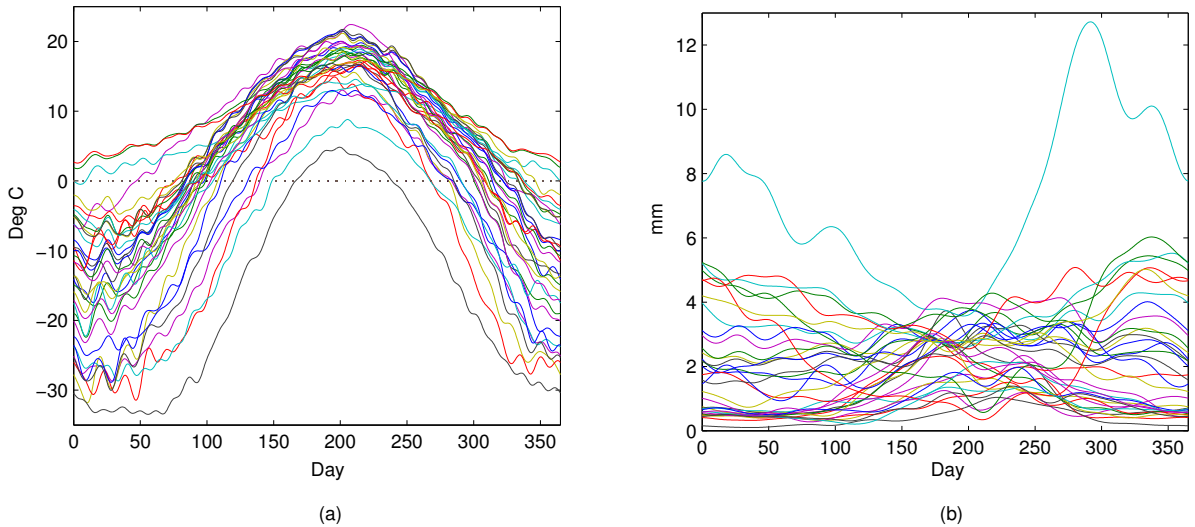


Fig. – Daily weather data for 35 Canadian station. (a) Temperature. (b) Precipitation.

- Predict the complete log daily precipitation profile of a weather station from information on temperature curves

Applications - Meteorology

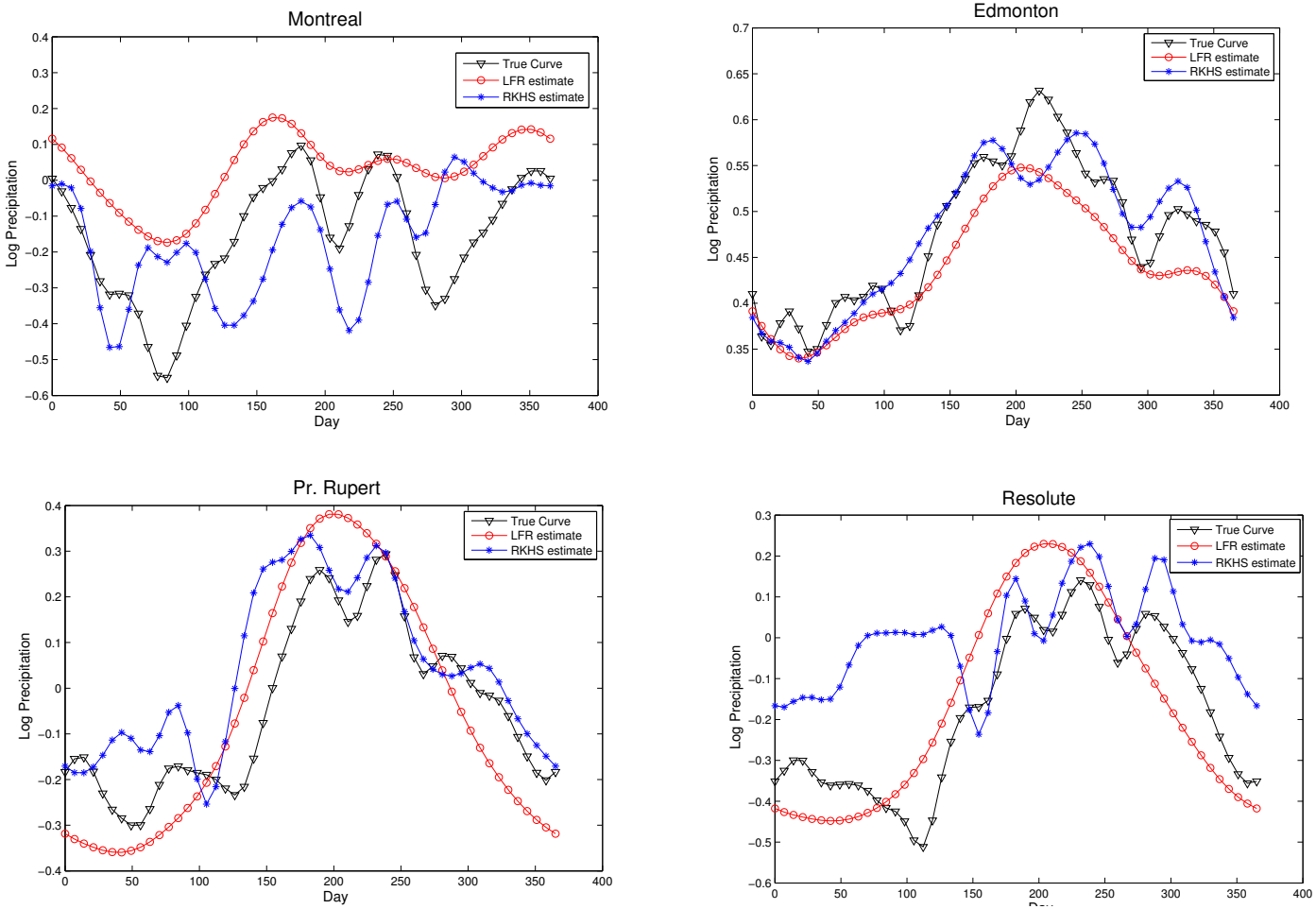


Fig. – True Curve (triangular mark), LFR prediction (circle mark) and RKHS prediction (star mark) of log precipitation for four weather station.

Other Applications - Physiology

- Lip-EMG data

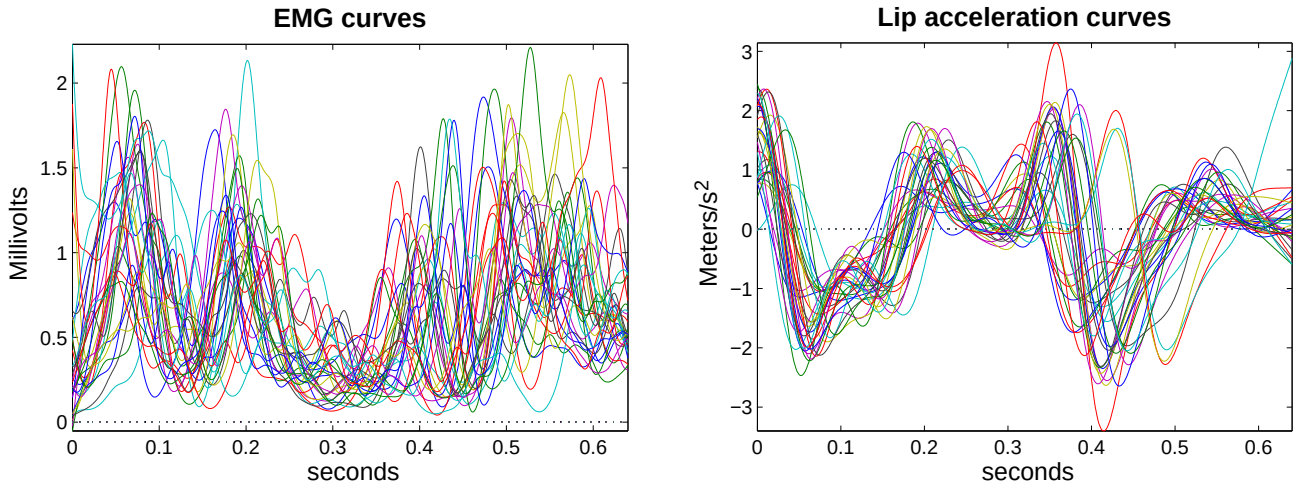


Fig. – EMG and lip acceleration curves. The left panel displays EMG recordings from a facial muscle that depresses the lower lip, the depressor labii inferior. The right panel shows the accelerations of the center of the lower lip of a speaker pronouncing the syllable "bob" for 32 replications.

- Study the dependence of the acceleration of the lower lip in speech on neural activity, as measured by electromyographical recordings (EMG)

Applications - Physiology

- The residual sum of squares error (RSSE)

$$RSSE = \int \sum_i \{y_i(t) - \widehat{y}_i(t)\}^2 dt$$

Tab. – Evaluation of the prediction of lip acceleration curves from EMG curves

Estimation method	RSSE
FLR - B-spline	793.44
RKHS - Multiplication operator	766.16
RKHS - Hilbert-Schmidt integral operator	715.37

Structured outputs

Reference : Bakir et al. (2007)
Predicting Structured Data. MIT Press.

Structured output learning - using Kernels

- **Regression :** output kernels (Weston et al., NIPS 2002 ; Cortes et al., ICML 2005 ; d'Alché-Buc et al., ICML 2011)

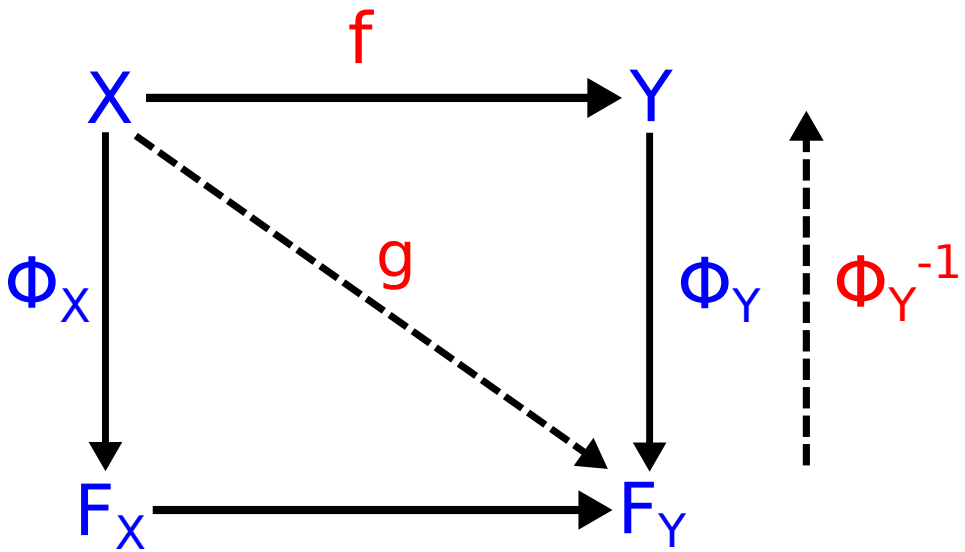
  $\rightarrow \; l(y_1, y_2)$

- Classification : Joint kernels (Taskar et al., NIPS 2003 ; Tsochantaridis, JMLR 2005)

  $\rightarrow \; J((x_1, x_2), (y_1, y_2)) = \langle \mathbf{\Phi}(x_1, y_1) \mathbf{\Phi}(x_2, y_2) \rangle$

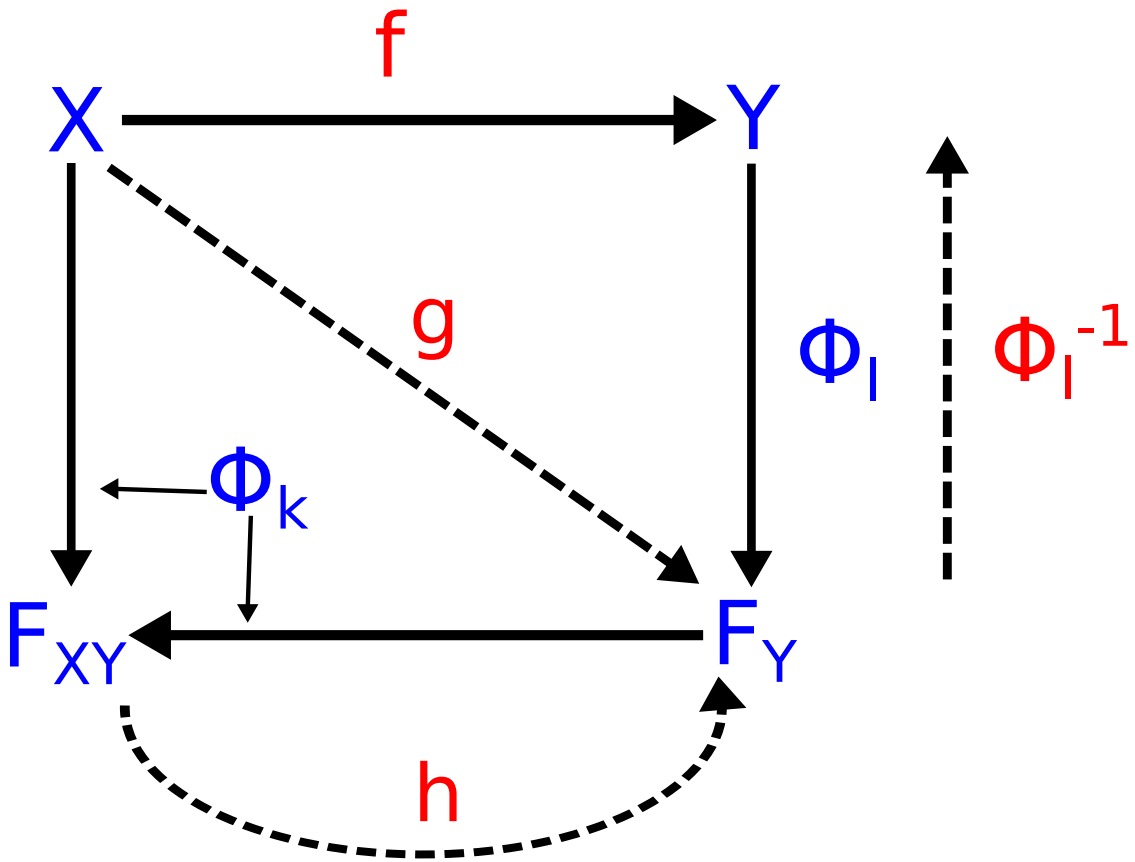
Structured output learning - KDE

- KDE - Kernel Dependency Estimation (Weston et al., 2002)



- Problems
  - $\rightarrow$ $\mathcal{F}_\mathcal{Y}$ Finite dimension
  - $\rightarrow$ Linear mapping h
  - $\rightarrow$ Independents outputs

Structured output learning - Advanced KDE



Advanced KDE - Optimization problem

- **1st step** : High dimensional output regression

  $\rightarrow \arg\min_{g \in \mathcal{F}_\mathcal{X}} \sum_{i=1}^{n} \|g(x_i) - \Phi_l(y_i)\|^2_{\mathcal{F}_\mathcal{Y}} + \lambda \|g\|^2$

  $\rightarrow h(.) = \sum_{i=1}^{n} K(., x_i)\psi_i \ , \quad \psi_i \in \mathcal{F}_\mathcal{Y}$

  $\rightarrow$ Solution : $\mathbf{\Psi} = (\mathbf{K} + \lambda I)^{-1}\mathbf{\Phi_l}$

- **2nd step** : Pre-image problem

  $\rightarrow$ Prediction : $f(x) = \arg\min_{y \in \mathcal{Y}} \|g(x) - \Phi_l(y)\|^2$

  $\rightarrow$ Solving : $\arg\min_{y \in \mathcal{Y}} \|K_x(\mathbf{K} + \lambda I)^{-1}\mathbf{\Phi_l} - \Phi_l(y)\|^2$

Advanced KDE - Pre-image

- Solving : $\arg\min_{y\in\mathcal{Y}} \|K_x(\mathbf{K}+\lambda I)^{-1}\mathbf{\Phi_1} - \Phi_l(y)\|^2$

  $\rightarrow$ Problem : kernel features **? ? ?**

  $\rightarrow$ Generalized kernel trick : $\langle T\Phi_l(y_1), \Phi_l(y_2)\rangle_{\mathcal{F}_\mathcal{Y}} = [Tl(y_1,.)](y_2)$

  $\rightarrow$ $T = id$: usual kernel trick $\langle \Phi_l(y_1), \Phi_l(y_2)\rangle_{\mathcal{F}_\mathcal{Y}} = l(y_1,y_2)$

  $\rightarrow$ $\arg\min_{y\in\mathcal{Y}} l(y,y) - 2\sum_i [(K_x(\mathbf{K}+\lambda I)^{-1})_i l(y_i,.)](y)$

Advanced KDE - Covariance operator

- **Covariance** operator in a RKHS (K. Fukumizu, L. Song, A. Gretton, NIPS 2011) : probabilistic framework

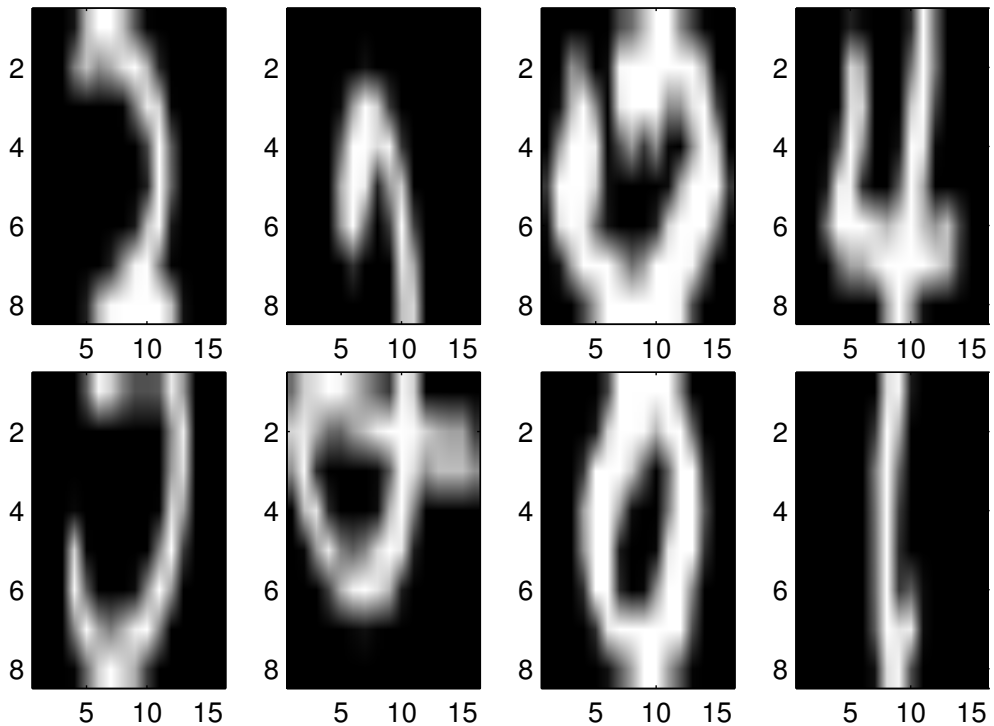
- $T_{yy} = N^{-1} \sum_{i=1}^{N} l(\cdot, y_i) \otimes l(\cdot, y_i)$ where $(y_1 \otimes y_2)h = \langle y_2, h \rangle y_1$

- **Proposition 1 :** The Gram matrix expression of the pre-image :

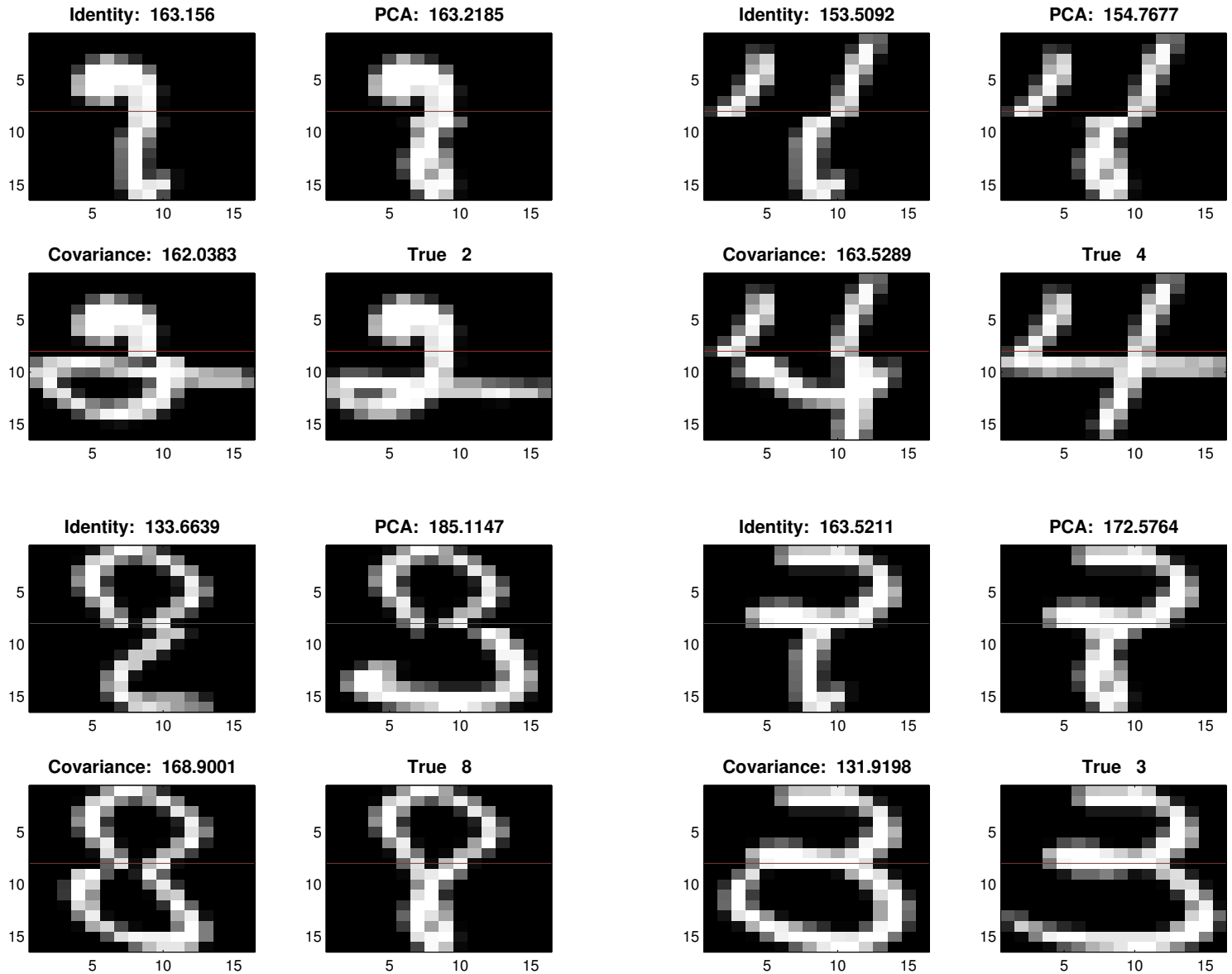
$$\arg\min_{y \in \mathcal{Y}} l(y,y) - 2L^T(y)[K(x) \otimes L][K \otimes L + \lambda I]^{-1} vec(Id)$$

Experiments - Image Reconstruction

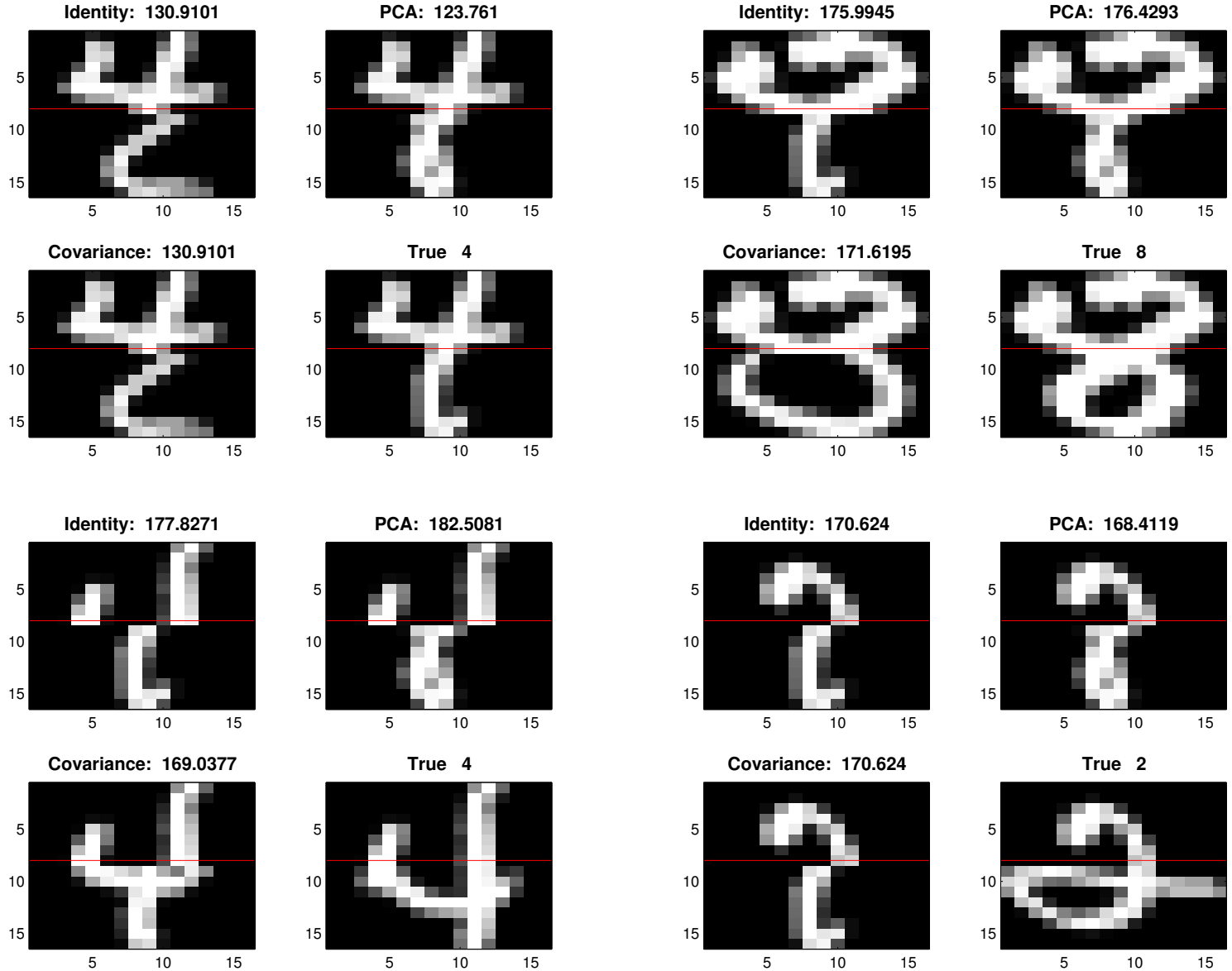
- Image reconstruction - USPS postal digit database



Experiments - Image Reconstruction

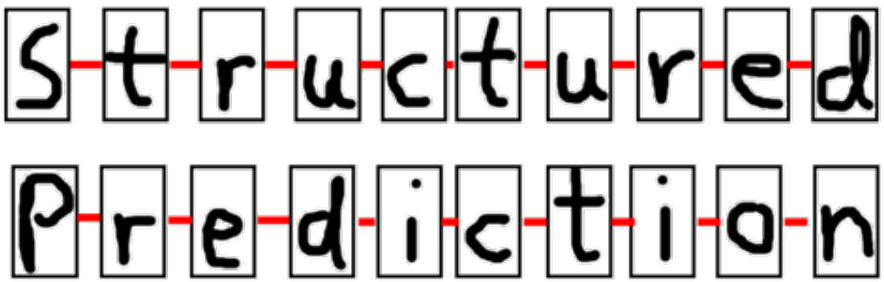


Experiments - Image Reconstruction



Experiments - Optical character recognition

- OCR database



- predicting a word from the sequence of pixel-based images of its handwritten characters.

Experiments - Face-to-Face Mapping

- Face database

- mapping the rotated view of a face to the plain expression (frontal view) of the same face.

PART III

From Learning with Kernels to Learning Kernels

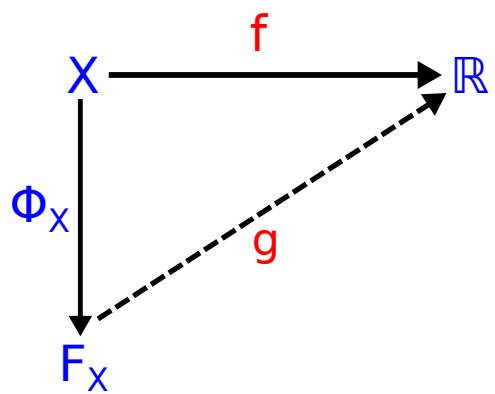
MKL : Multiple Kernel Learning

Kernel - Learning *with* kernels

$$y_i = f(x_i) + \epsilon_i \quad ; \quad y_i \in \mathbb{R}$$

2 Perspectives

- RKHS theory
  - $\longrightarrow$ Mercer theorem : integral operator + positive kernel
  - $\longrightarrow$ reproducing property : $\langle f, k(x, \cdot) \rangle = f(x)$
  - $\longrightarrow$ representer theorem : $f(\cdot) = \sum_i \alpha_i k(x_i, \cdot)$; $\alpha_i \in \mathbb{R}$
- Feature space
  - $\longrightarrow$ nonlinear in input space — linear in Feature space
  - $\longrightarrow$ projecting data into a Feature space (high dimension)
  - $\longrightarrow$ kernel trick $\langle \Phi(x_1), \Phi(x_2) \rangle = k(x_1, x_2)$



Kernel - Learning ~~with~~ kernels

3 Perspectives

- MKL : multiple kernel learning
- Hyperkernels : kernel on kernels
- via regularization
- ...

MKL $\Rightarrow$ linear combination $=$ weighted sum of kernels

- $k(\cdot,\cdot) = \sum_j d_j k_j(\cdot,\cdot)$
- $d = [d_1, \cdots, d_M]$; $\{d : \forall j, d_j \geq 0, \sum_j d_j^r \leq 1\}$; $1 \leq r \leq \infty$
- $r = 1$ norm 1 (sparse) ; $r = 2$ norm 2 (non sparse)

Kernel - Learning *with* operator-valued kernels

$$y_i = f(x_i) + \epsilon_i \quad ; \quad y_i \in \mathcal{Y}$$

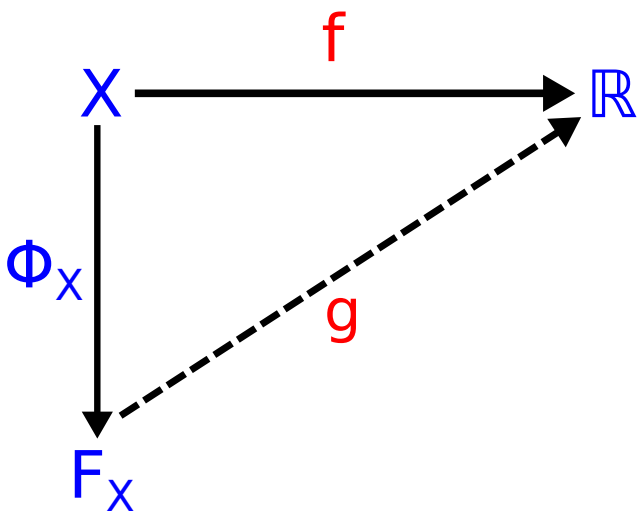
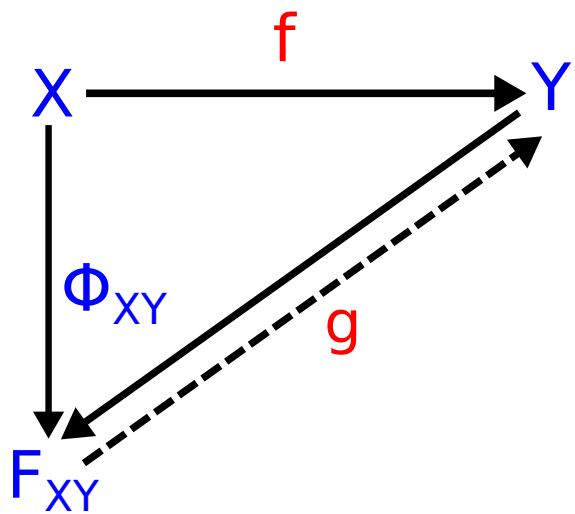
$\longrightarrow$ multi-task learning : $\mathcal{Y} = \mathbb{R}^p$

$\longrightarrow$ functional regression : $\mathcal{Y} = L^2([a,b])$

2 Perspectives

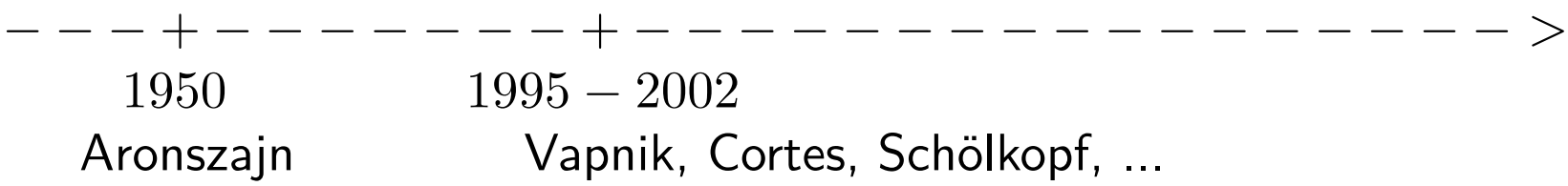
- RKHS theory
  - $\longrightarrow$ reproducing property : $\langle f, K(x,\cdot)w \rangle = \langle f(x), w \rangle_{\mathcal{Y}}$
  - $\longrightarrow$ representer theorem : $f(\cdot) = \sum_i K(x_i,\cdot)\alpha_i \; ; \; \alpha_i \in \mathcal{Y}$
- Feature space
  - $\longrightarrow$ nonlinear multi-task methods
  - $\longrightarrow$ kernel trick $\langle \Phi(x_1,w_1), \Phi(x_2,w_2) \rangle = \langle K(x_1,x_2)w_1, w_2 \rangle_{\mathcal{Y}}$

Kernel - Learning *with* operator-valued kernels

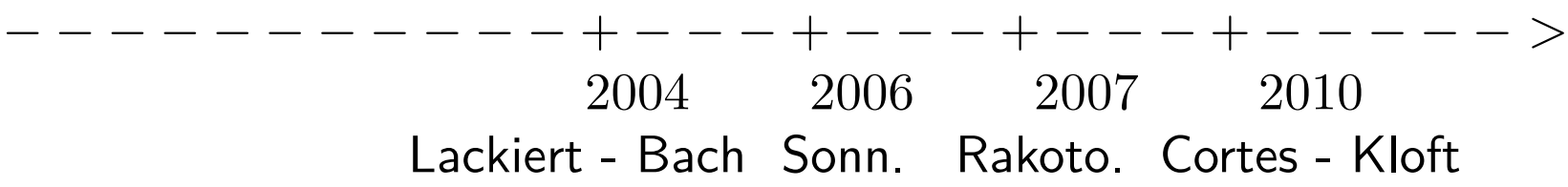
Scalar-valued	Operator-valued
	
$\langle f, k(x,\cdot)\rangle = f(x)$	$\langle f, K(x,\cdot)w\rangle = \langle f(x), w\rangle_{\mathcal{Y}}$
$\langle \Phi(x_1), \Phi(x_2)\rangle = k(x_1,x_2)$	$\langle \Phi(x_1,w_1), \Phi(x_2,w_2)\rangle = \langle K(x_1,x_2)w_1, w_2\rangle_{\mathcal{Y}}$
$f(\cdot) = \sum_i \alpha_i k(x_i,\cdot)\,;\ \alpha_i \in \mathbb{R}$	$f(\cdot) = \sum_i K(x_i,\cdot)\alpha_i\,;\ \alpha_i \in \mathcal{Y}$

Kernel - Learning ~~with~~ operator-valued kernels

$\longrightarrow$ Learning with kernels



$\longrightarrow$ Learning ~~with~~ kernels



$\longrightarrow$ Learning with operator-valued kernels

$\longrightarrow$ **What about learning ~~with~~ operator-valued kernels ?**

- **Our solution : MovKL = MKL + OVK**

Outline

- **MovKL**
  - → Formulation
  - → Algorithm
- **Block-operator kernel matrices**
  - → Multiple operator-valued kernel matrices
  - → Gauss-Seidel Algorithm
- **Experiment - BCI dataset**
  - → Finger state prediction
  - → Finger movement prediction

MovKL - Formulation

$$\min_{d\in\mathcal{D}} \min_{f_k\in\mathcal{F}_k} \sum_{k=1}^{M} \frac{\|f_k\|^2_{\mathcal{F}_k}}{2d_k} + \frac{1}{2n\lambda}\sum_{i=1}^{n} \|\xi_i\|^2_{\mathcal{G}_y}$$
$$\text{with } \xi_i = y_i - \textstyle\sum_{k=1}^{M} f_k(x_i),$$

- $f(\cdot)$ is sum of M functions $\{f_k(\cdot)\}_{k=1}^{M}$
- $d = [d_1, \cdots, d_M]$, $\mathcal{D} = \{d : \forall k, d_k \geq 0, \sum_k d_k^r \leq 1\}$, $1 \leq r \leq \infty$

Dualization

$$\min_{d\in\mathcal{D}} \max_{\alpha\in\mathcal{G}_y^n} -\frac{n\lambda}{2}\|\alpha\|^2_{\mathcal{G}_y^n} - \frac{1}{2}\langle \mathbf{K}\alpha, \alpha\rangle_{\mathcal{G}_y^n} + \langle \alpha, y\rangle_{\mathcal{G}_y^n}$$

MovKL - Solution

- alternating optimization
- with $\{d_k\}$ fixed

  $\longrightarrow (\mathbf{K} + \lambda I)\alpha = y$

  $\longrightarrow \mathbf{K} = \sum_{k=1}^{M} d_k \mathbf{K}_k$

- with $\{f_k\}$ fixed

  $$\longrightarrow \min_{d \in \mathcal{D}} \sum_{k=1}^{M} \frac{\|f_k\|^2_{\mathcal{F}_k}}{d_k}$$

  $$\longrightarrow d_k = \frac{\|f_k\|^{\frac{2}{r+1}}}{(\sum_k \|f_k\|^{\frac{2r}{r+1}})^{1/r}}$$

MovKL - Algorithm

Algorithm 1 ℓ_r-norm MovKL

Input $\mathbf{K}_k$ for $k = 1, \ldots, M$

$d_k^1 \longleftarrow \dfrac{1}{M}$ for $k = 1, \ldots, M$

$\alpha \longleftarrow 0$

for $t = 1, 2, \ldots$ **do**

  $\alpha' \longleftarrow \alpha$

  $\mathbf{K} \longleftarrow \sum_k d_k^t \mathbf{K}_k$

  $\alpha \longleftarrow$ solution of $(\mathbf{K} + \lambda I)\alpha = y$

  if $\|\alpha - \alpha'\| < \epsilon$ **then**

    break

  end if

  $d_k^{t+1} \longleftarrow \dfrac{\|f_k\|^{\frac{2}{r+1}}}{(\sum_k \|f_k\|^{\frac{2r}{r+1}})^{1/r}}$ for $k = 1, \ldots, M$

end for

MovKL - Problem

- $\alpha \longleftarrow$ solution of $(\mathbf{K} + \lambda I)\alpha = y$
- $\mathbf{K}$: sum of block operator kernel matrices
- block operator matrix $=$ block $/$ block $=$ operator
- operator $=$ matrix in infinite dimension
- separable operator-valued kernel $K(w, z) = G(w, z)T$

MovKL - Solution - Kernel combination

- $\alpha \longleftarrow$ solution of $(\mathbf{K} + \lambda I)\alpha = y$

- case 1 : $K(w,z) = \sum_{k=1}^{M} d_k G_k(w,z) T$

  $\longrightarrow \mathbf{K} = \mathbf{G} \otimes T$

  $\longrightarrow \mathbf{G} = \sum_{k=1}^{M} d_k \mathbf{G}_k$

  $\longrightarrow \mathbf{K}^{-1} = \mathbf{G}^{-1} \otimes T^{-1}$

- case 2 : K(w,z) $= \sum_{k=1}^{M} d_k G_k(w,z) T_k$

  $\longrightarrow$ Gauss-Seidel alorithm

  $\longrightarrow$ Iterative algorithm

MovKL - Solution - Gauss-Seidel

- $\alpha \longleftarrow$ solution of $(\mathbf{K} + \lambda I)\alpha = y$

Algorithm 2 Gauss-Seidel Method

---

choose an initial vector of functions $\alpha^{(0)}$

repeat

  for $i = 1, 2, \ldots, n$

    $\alpha_i^{(t)} \longleftarrow$ sol. of : $[K(x_i, x_i) + \lambda I]\alpha_i^{(t)} = s_i$

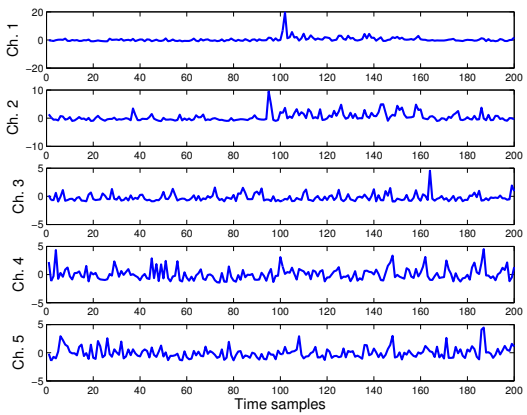
  end for

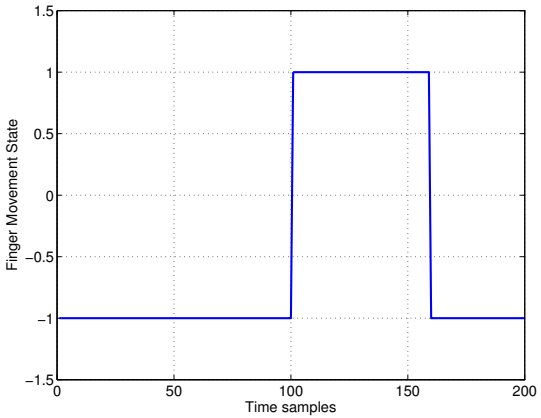
until convergence

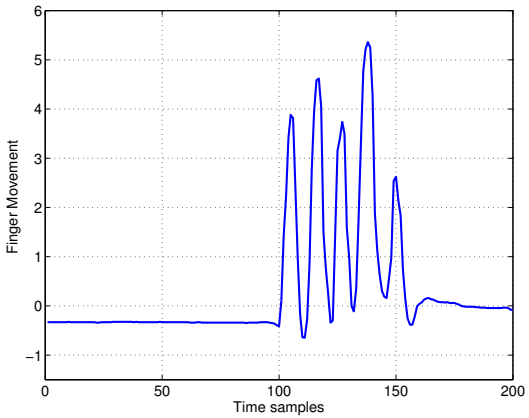
- $[K(x_i, x_i)+\lambda I]\alpha_i^{(t)} = y_i - \sum_{j=1}^{i-1} K(x_i, x_j)\alpha_j^{(t)} - \sum_{j=i+1}^{n} K(x_i, x_j)\alpha_j^{(t-1)}$

- inverse : sum block matrices $\longrightarrow$ sum matrices $\longrightarrow$ matrices
  GS1 GS2

BCI - Functional regression







- Supervised learning problem when both covariates and responses are real functions rather than scalars or finite dimensional vectors

  $\longrightarrow$ functional regression with functional responses

- Nonparametric operator estimation

  $\longrightarrow$ function-valued RKHS and operator-valued kernels

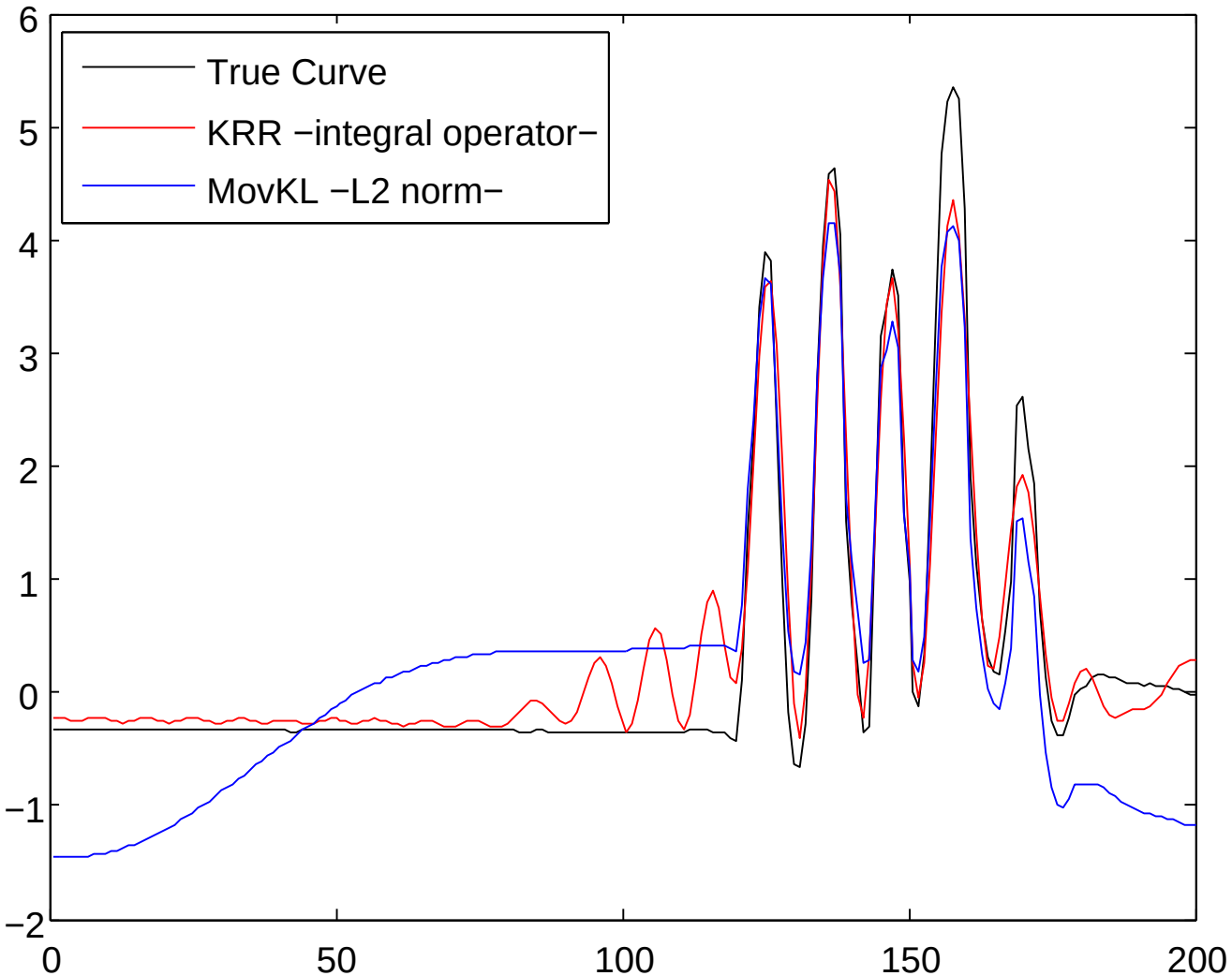
BCI - Results

- kernels : Gaussian (5 bandwidths) + polynomial (1 to 3 degree)
- operators : identity + multiplication (e^{-t^2}) + integral ($e^{-|t-s|}$)

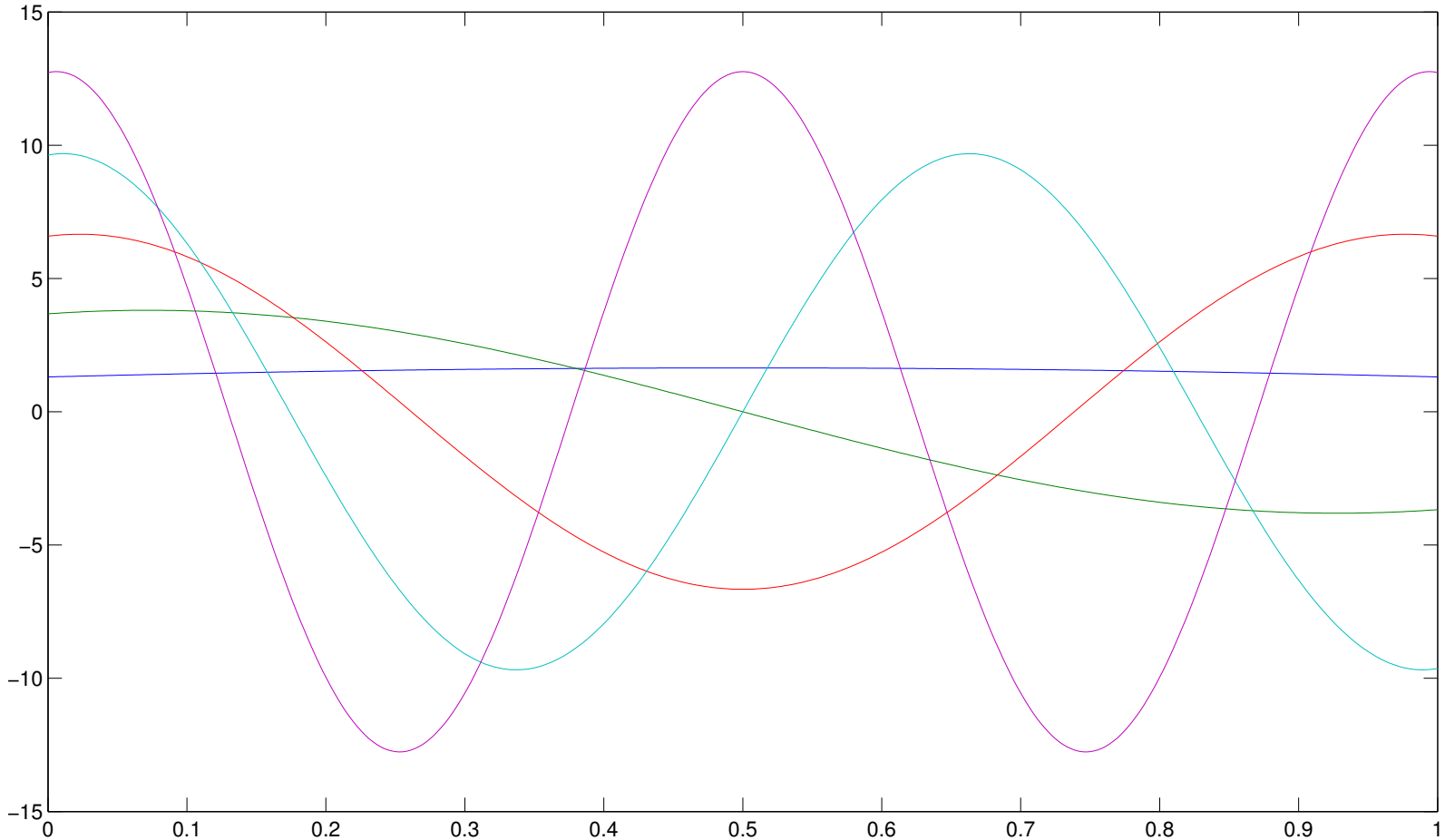
Algorithm	RSSE	LCR(%)
KRR - scalar-valued -	68.32	72.91
KRR - functional response -	49.40	80.20
MovKL - ℓ_∞ norm -	45.44	81.34
MovKL - ℓ_1 norm -	48.12	80.66
MovKL - ℓ_2 norm -	**39.36**	**84.72**

Algorithm	RSSE
KRR - scalar-valued -	88.21
KRR - functional response -	79.86
MovKL - ℓ_∞ norm -	76.52
MovKL - ℓ_1 norm -	78.24
MovKL - ℓ_2 norm -	**75.15**

BCI - Results


True Curve
KRR −integral operator−
MovKL −L2 norm−
6
5
4
3
2
1
0
−1
−2
0
50
100
150
200

BCI - Results



BCI - Results